

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR**



**FACULTAD DE INGENIERÍA**  
**MAESTRÍA EN REDES DE COMUNICACIÓN**

**INFORME FINAL CASO DE ESTUDIO PARA UNIDAD DE TITULACIÓN  
ESPECIAL**

**TEMA:**

**“PROYECTO DE IMPLEMENTACIÓN DE GEOLOCALIZACIÓN EN EL  
MÓDULO DE VENTAS DE MICROSOFT DYNAMICS AX MEDIANTE  
EL DISEÑO Y CONSTRUCCIÓN DE UNA APLICACIÓN MÓVIL”**

***JOSE GUSTAVO LEORO RUALES***

Quito – 2015

## Tabla de contenido

TABLA DE FLUJOGRAMAS.....	3
TABLA DE GRÁFICOS.....	3
TABLA DE ILUSTRACIONES .....	3
LISTADO DE TABLAS .....	4
I. Introducción .....	1
II. Justificación .....	2
III. Antecedentes .....	3
IV. Objetivos .....	3
V. Desarrollo Caso de Estudio.....	5
5.1 Ciclo del Proceso .....	5
5.2 Módulo de Administración de Rutas en el ERP .....	6
5.3 Arquitectura e Infraestructura Microsoft Dynamics Ax .....	18
5.4 Esquema de Integración .....	22
5.5 Consideraciones a nivel de seguridad .....	24
5.6 Servicios Web personalizados en Microsoft Dynamics Ax .....	25
5.7 Esquema de intercambio de información .....	28
5.8 Estructura de documento de Servicios .....	32
5.9 API de Rutas de Google .....	44
5.10 Core Location Framework iOS.....	48
5.11 Aplicación Móvil .....	52
5.12 Funcionamiento Off-Line.....	65
5.13 Conclusiones y Recomendaciones.....	68
CONCLUSIONES .....	68
RECOMENDACIONES .....	70
VI. Bibliografía: .....	71
VII. Anexos.....	73
ANEXO 1. DIAGRAMA ENTIDAD RELACIÓN:.....	73
Anexo 2. DICCIONARIO DE DATOS .....	75
Anexo 3. Diagrama de clases iOS (Core Data Model).....	78

## TABLA DE FLUJOGRAMAS

Flujograma 1.- Ciclo de procesos de ventas .....	6
--	---

## TABLA DE GRÁFICOS

Gráfico 1 Administración de Usuarios Móviles .....	8
Gráfico 2.- Administración de Usuarios Móviles .....	9
Gráfico 3.- Información Adicional de Vendedores .....	10
Gráfico 4 Datos demográficos de ventas de clientes .....	11
Gráfico 5 Información geográfica de clientes .....	12
Gráfico 6 Coordenadas geográficas de Clientes.....	13
Gráfico 7 Rutas de Venta.....	14
Gráfico 8 Programación de Ruta de Venta.....	15
Gráfico 9 Programación de Ruta de Venta.....	15
Gráfico 10 Generación masiva de Rutas .....	16
Gráfico 11 Visualización detallada de rutas .....	17
Gráfico 12 Topología básica - Microsoft Dynamics Ax .....	18
Gráfico 13 Topología escalable - Microsoft Dynamics Ax .....	19
Gráfico 14 Arquitectura del Marco de Integración de Aplicación - Microsoft Dynamics Ax .....	23
Gráfico 15 Componentes de la notación JSON .....	31
Gráfico 16.- Componentes de la notación JSON .....	31
Gráfico 17 Componentes de la notación JSON .....	31
Gráfico 18 Inicio de Sesión .....	52
Gráfico 19 Ruta de Clientes.....	53
Gráfico 20 Ruta Sugerida - Aplicación Móvil .....	54
Gráfico 21 Generación de Orden de Venta .....	54
Gráfico 22 Órdenes de venta generadas.....	55
Gráfico 23 Crear Orden de Venta .....	55
Gráfico 24 Creación de Orden de Venta .....	56
Gráfico 25 Creación de Orden de Venta .....	56
Gráfico 26 Detalle de Orden de Venta .....	57
Gráfico 27 Factura de Orden de Venta .....	58
Gráfico 28 Zonas de Venta .....	59
Gráfico 29 Listado de Vendedores .....	59
Gráfico 30 Monitoreo de actividades - Vendedor.....	60
Gráfico 31 Registro de Sucursales .....	61
Gráfico 32 Listado de Clientes.....	62
Gráfico 33 Listado de Sucursales por Cliente .....	63
Gráfico 34 Ingreso o edición de sucursales.....	64
Gráfico 35 Sincronización de Pedidos de Venta.....	66
Gráfico 36 Pedidos Pendientes de Sincronizar.....	67
Gráfico 37 Diagrama Entidad Relación - Módulo de gestión de rutas .....	74
Gráfico 38 Diagrama de clases de la aplicación móvil .....	78

## TABLA DE ILUSTRACIONES

Ilustración 1.- Recepción de producto .....	7
---	---

## LISTADO DE TABLAS

Tabla 1.- Configuración Usuarios Móviles.....	8
Tabla 2.- Configuración adicional de vendedores.....	10
Tabla 3 Administraciones zonales (Zonas de Venta).....	11
Tabla 4 Información geográfica de clientes .....	12
Tabla 5 Indicador de actividad del vendedor .....	60
Tabla 6 Registro de Sucursales .....	64
Tabla 7 Diccionario de datos - Usuarios Móviles .....	75
Tabla 8 Diccionario de Datos - Información adicional de vendedores.....	75
Tabla 9 Diccionario de Datos - Información general de Ruta.....	75
Tabla 10 Diccionario de Datos - Plantilla de Rutas de Venta .....	76
Tabla 11 Información de Detalle de Ruta.....	76
Tabla 12 Diccionario de Datos - Información de Orden de Venta .....	77
Tabla 13 Diccionario de Datos - Información de Facturas de Orden de Venta .....	77

## I. Introducción

Actualmente con el crecimiento significativo de diversas aplicaciones para plataformas móviles (dispositivos celulares) las soluciones basadas en el uso de geolocalización se han vuelto más populares y aplicables en el día a día de las organizaciones con el fin de permitir una administración más óptima de sus recursos.

Analizando la problemática de algunas empresas, se evidenció la importancia de organizar y supervisar a los equipos de ventas mediante la construcción de una aplicación para dispositivos móviles (celulares), que aproveche las características de su GPS integrado y permita ofrecer información relevante de las rutas de ventas y toma de pedidos de venta que son actualmente administradas desde el sistema central de una organización (ERP)

Se entiende por geolocalización la ubicación satelital de una entidad, en nuestro caso el vendedor que transmitirá su posición geográfica a través del dispositivo móvil.

ERP, Por sus siglas en ingles “Enterprise Resource Planning”, es un sistema integrado de información que permite planificar los recursos empresariales.

Con la aplicación de un sistema tecnológico ERP se pretende mejorar los procesos de planificación y control de los recursos, que posee una empresa.

Microsoft Dynamics AX, es el ERP comercializado por la compañía Microsoft desde el año 2002, siendo su nombre original Axapta.

Existen varias versiones de este ERP, sin embargo la versión utilizada para el es Microsoft Dynamics Ax 2012 R2.

## II. Justificación

Considerando que la tendencia y enfoque de las empresas se centra en la búsqueda e implementación de soluciones móviles, se vio la necesidad de desarrollar una aplicación que se pueda integrar al trabajo diario del personal de una organización, adicionalmente se evidenció una escasez en el mercado de soluciones que permitan combinar el uso de estas tecnologías con sistemas de posicionamiento global.

Microsoft Dynamics Ax es el ERP que poseen algunas grandes empresas en el país, una de ellas Confiteca, compañía dedicada a la fabricación y distribución de productos.

Las ventas es el punto más crítico en este tipo de empresas, es importante una buena organización y sobre todo planificación de la demanda que existe en el mercado acompañada de un control de actividades sobre cada vendedor móvil

En esta empresa actualmente existe una integración entre su ERP y los dispositivos hand-held y PDA, sin embargo esta combinación únicamente permite desarrollar las actividades comerciales más no de análisis, sumado a esto que el proceso de sincronización de la información entre el ERP y estos dispositivos no es en línea, por lo tanto cada vendedor requiere acercarse a puntos específicos destinados para esta actividad.

Por lo tanto se ha decidido desarrollar una aplicación móvil, integrada a Microsoft Dynamics AX mediante el uso de servicios web, que permita enviar desde el dispositivo móvil hacia el ERP información precisa respecto a los pedidos de venta procesados por cada vendedor junto con la posición geográfica de las ventas efectuadas por cada vendedor para su posterior análisis.

Por otro lado, Confiteca actualmente realiza el levantamiento de información de coordenadas de clientes de manera manual, sin embargo con el desarrollo esta aplicación la toma de información será automática y en línea, a través del dispositivo móvil y actualizado en el ERP mediante un servicio web.

### III. Antecedentes

En Quito, la empresa Confiteca, cuando inició la implementación de Microsoft Dynamics AX, desarrolló un módulo de ventas basado en el concepto de rutas de ventas, esta información es trasladada mediante una interfaz a cada uno de los dispositivos móviles hand-held y PDA

La información almacenada en estos dispositivos genera datos de entrada para poder realizar las ventas asignadas al personal, permitiendo a los vendedores despachar los productos y armar los pedidos de venta. Toda esta información retorna al sistema ERP también mediante una interfaz.

Con el pasar de los años, surge la necesidad de poder utilizar de mejor manera toda la información que es almacenada en el dispositivo móvil y no se traslada al ERP, como por ejemplo la ubicación exacta de cada uno los puntos visitados, ventas efectuadas; esto para poder obtener una estadística de ventas por zona más exacta y en línea, y más importante aún, tener un control de actividades sobre cada vendedor.

### IV. Objetivos

#### ***Objetivo General:***

Obtener información respecto a la ubicación exacta que posee un cliente de la organización, en el momento que se realiza la venta, para almacenar y actualizar en el ERP Microsoft Dynamics AX, y con esto poder realizar una planificación de la demanda por zonas y sitios específicos y tener un control más efectivo de la situación de cada vendedor.

Para conseguir este objetivo inicial, se plantea la construcción de una demo sobre la plataforma iOS 8.0 cuyo modelo será extendido en el corto plazo al resto de plataformas de dispositivos móviles existentes en el mercado.

**Objetivos Específicos:**

1. Desarrollar en Microsoft Dynamics Ax un módulo de administración y gestión de rutas de ventas basadas en la experiencia de Confiteca y que aplique en su totalidad a empresas similares dedicadas a la distribución masiva de productos de diversa variedad.
2. Desarrollar una aplicación para dispositivos móviles que permita gestionar los pedidos de venta y obtener la ubicación exacta con coordenadas geográficas, del lugar en el cual se realiza dicha venta; así como la visualización de rutas sugeridas hacia cada uno de los clientes que forman parte de la ruta.
3. Desarrollar una interfaz de doble vía mediante el uso de servicios web, que permita enviar desde el ERP hacia el dispositivo móvil toda la información necesaria para generar los pedidos de venta de cada cliente de la organización y retornar hacia el ERP la venta en firme junto con la posición geográfica.
4. Desarrollar dentro de la aplicación móvil un módulo de consultas que permita el análisis de los datos obtenidos utilizando geolocalización y el apoyo visual de mapas para una mejor interpretación de estos datos.



## **V. Desarrollo Caso de Estudio**

### **5.1 Ciclo del Proceso**

El proceso inicia con la recepción del producto por parte del vendedor. La recepción implica que el vendedor acude físicamente a los centros de abastecimiento de la organización, y se aprovisiona del inventario disponible en su almacén “móvil”, para ser más precisos el almacén “móvil” dentro del ERP corresponde a una bodega de inventario y físicamente representa el camión, de propiedad del vendedor o de la organización.

El proceso de transferencia de inventario desde las distintas bodegas de la organización a cada uno de los almacenes móviles se los realiza desde el sistema central ERP y es ejecutado por usuarios exclusivos de la aplicación Microsoft Dynamics Ax.

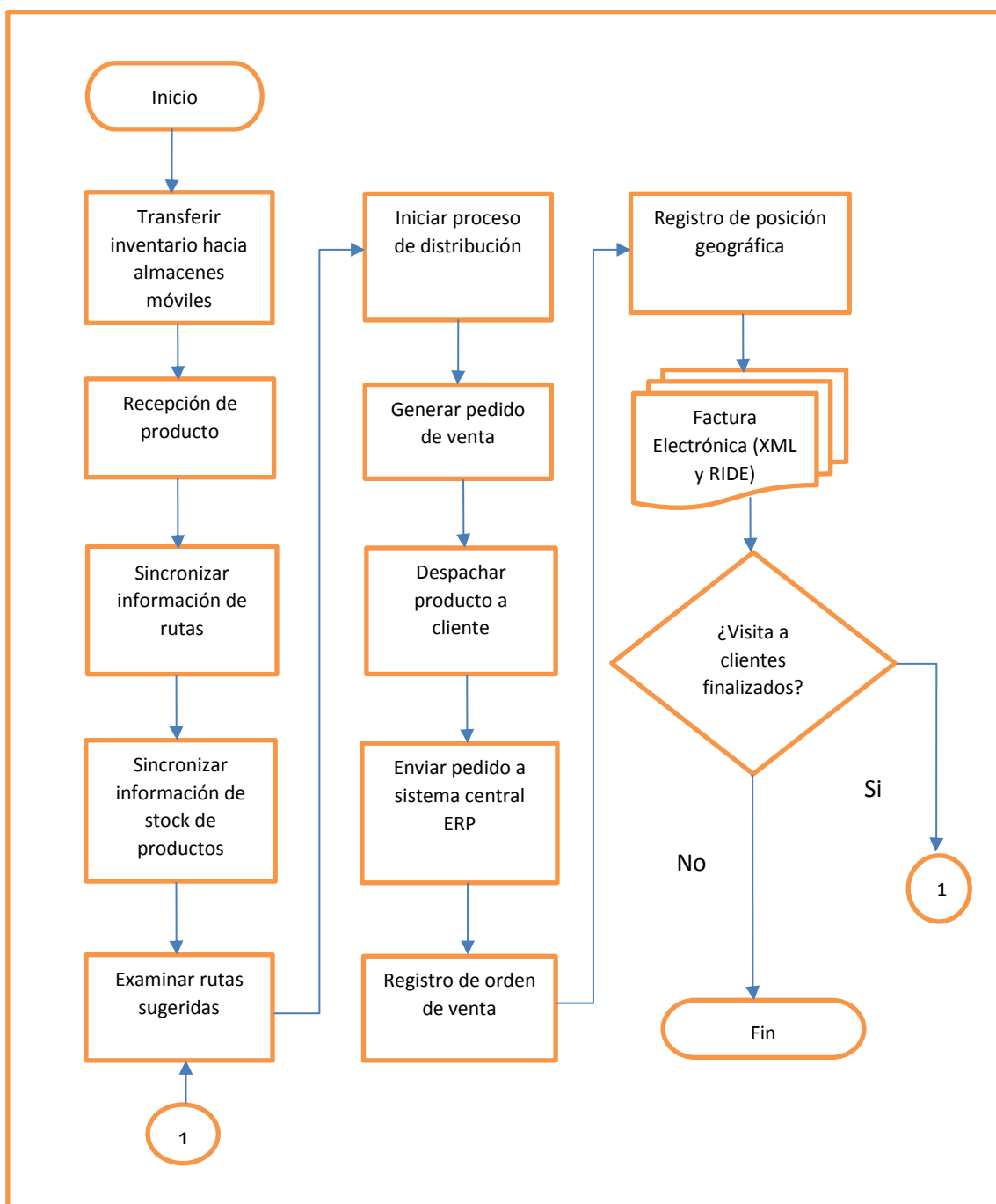
Por otra parte, el vendedor con su dispositivo móvil (Tablet o teléfono inteligente) inicia sesión en la aplicación móvil, esta acción internamente invoca a un servicio web que realiza la carga de toda la programación de visitas planificadas para un día específico conjuntamente con el stock de inventario disponible en el almacén móvil del vendedor. Este proceso de carga inicial se lo puede realizar usando la red móvil 3G/4G o una red inalámbrica.

La aplicación sugiere, en función su ubicación actual, el orden de los clientes a visitar comenzando por el más cercano.

La visita a cada cliente implica la creación de un pedido de venta, mejor conocido en el mundo de los ERP como orden de venta.

Una orden de venta se encuentra asociada a un cliente y en su detalle se incluye el listado de los artículos a despachar. El proceso termina con la entrega del pedido y la facturación de la orden de venta, esto significa que la aplicación móvil envía los datos del pedido de venta hacia el sistema central ERP, la aplicación móvil realiza el consumo de un servicio web y este se encarga de contabilizar la orden de venta, genera y autorizar la factura, y finalmente realizar el descargo de inventario.

Flujograma 1.- Ciclo de procesos de ventas



Fuente: Gustavo Leoro

## 5.2 Módulo de Administración de Rutas en el ERP

De manera estándar el ERP no cuenta con un módulo de gestión y administración de rutas, es por ello que para llevar a cabo la propuesta planteada en el caso de estudio ha sido necesario el desarrollo de este nuevo módulo que tiene como premisa fundamental la asociación de un

vendedor con una bodega de inventario, y la asignación de rutas que corresponden a un conjunto de clientes a visitar en un día específico.

El objetivo central de esta propuesta, es permitir la transferencia de inventario desde la bodega o bodegas principales hacia cada una de las bodegas móviles asociadas a cada vendedor, con el fin de abastecerlo del inventario necesario para iniciar el proceso de distribución.

*Ilustración 1.-  
Recepción de  
producto*



**Fuente: Gustavo Leoro**

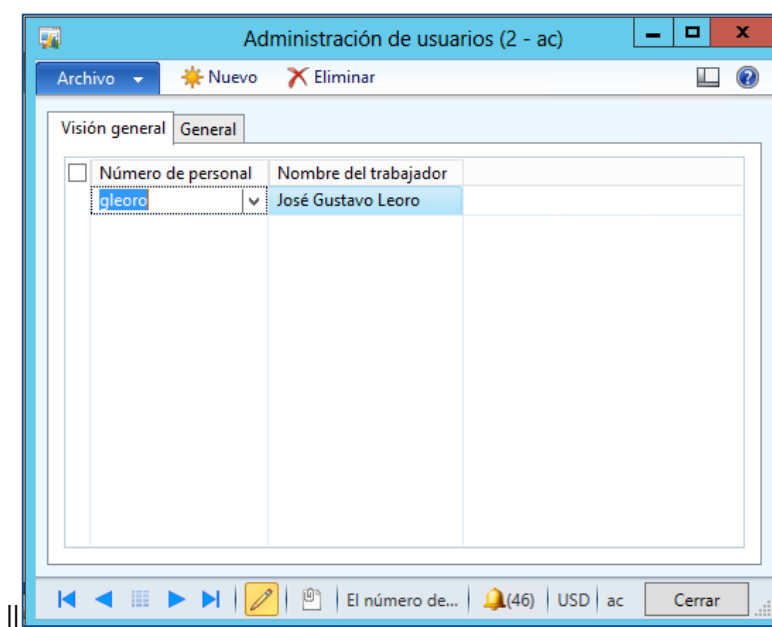
Es importante mencionar que la gestión y administración de los usuarios (vendedores) que harán uso de la aplicación móvil se lo realiza en el sistema central ERP.

Para efectos de nuestro caso de estudio se han definido dos perfiles:

- Vendedor: Este tipo de usuario tendrá únicamente acceso a la funcionalidad de generación y registro de pedidos de venta.
- Administrador: Este tipo de usuario además de la funcionalidad antes descrita, tendrá acceso a monitorear el avance de actividades y ventas efectivas de los vendedores a su cargo.

En el sistema ERP, se ha creado una nueva sesión para permitir el registro de los usuarios que accederán a la aplicación móvil.

*Gráfico 1 Administración de Usuarios Móviles*



Fuente: Dynamics Ax – Módulo de Gestión de Rutas

Como se muestra en el gráfico No. 1 es necesario completar la siguiente información:

*Tabla 1.- Configuración Usuarios Móviles*

Campo	Descripción
Número de Personal	Corresponde al código identificador de cada empleado dentro de la organización
Contraseña	Corresponde a la clave de acceso

	para la aplicación móvil. La contraseña se encuentra encriptada para proporcionar mayor nivel de seguridad
Perfil	Se han definido dos perfiles: Administrador y Vendedor

Fuente: Dynamics AX – Módulo de Gestión de Rutas

Gráfico 2.- Administración de Usuarios Móviles

Fuente: Dynamics Ax – Módulo de Gestión de Rutas

## VENDEDOR

Un vendedor dentro del sistema central ERP, corresponde a un empleado o trabajador de la organización. El módulo de recursos humanos de Microsoft Dynamics Ax se encarga de gestionar y administrar toda la información relacionada con cada empleado de la organización.

Se han realizado unos pequeños ajustes y modificaciones a este módulo para permitir el ingreso de la información necesaria que permitirá completar de forma adecuada el ciclo de toma de pedidos de venta, la cual se detalla a continuación:

Gráfico 3.- Información Adicional de Vendedores

Fuente: Dynamics AX – Módulo de Gestión de Rutas

Tabla 2.- Configuración adicional de vendedores

Campo	Detalle
Sitio	Un sitio es una agrupación de recursos empresariales como, por ejemplo, almacenes y unidades de producción, que comparten transacciones relacionadas entre sí, poseen una ubicación geográfica y pertenecen a una única entidad jurídica.
Almacén	Corresponde a una ubicación dentro del sitio donde se almacena cualquier tipo de inventario.
Establecimiento	Corresponde a cada una de las sucursales que tiene la registrada la entidad jurídica en el Servicio de Rentas Internas
Punto de Emisión	Corresponde a cada uno de los puntos en los cuales se realiza la impresión de comprobantes autorizados. Cada punto de emisión tiene asociado un número secuencial que corresponde al número de factura.

Fuente: Dynamics AX – Módulo de Gestión de Rutas

La información de sitio y almacén permite notificar a cada vendedor de su inventario disponible (stock).

La información de establecimiento y punto de emisión permite registrar la factura de venta de acuerdo a lo estipulado en el reglamento de comprobantes electrónicos del Servicio de Rentas Internas.

## CLIENTE

Un cliente dentro del sistema central ERP, corresponde a aquella persona natural o jurídica que realiza la transacción comercial denominada compra, el cual contiene una serie de atributos que permiten definirlo claramente, sin embargo pondremos énfasis únicamente en aquellos que competen a nuestro caso de estudio.

Uno de los atributos importantes y que es parte fundamental de la propuesta de este caso de estudio es la posibilidad de clasificar a un cliente en función de su ubicación dentro de un área geográfica. Este atributo se conoce como zona de ventas.

*Gráfico 4 Datos demográficos de ventas de clientes*

The screenshot shows the 'Empresas cliente' form in Dynamics AX. The title bar indicates 'Empresas cliente (1 - ac) - Código de cliente: 1765324498, GUSTAVO LEORO R.'. The form has a ribbon with tabs: Archivo, Cliente, Vender, Factura, Cobrar, Proyectos, Servicio, Mercado, Comercial, and General. The 'Cliente' tab is active, showing various icons for actions like Eliminar, Editar, Mantener, Nuevo, Cuentas, Transacciones, Saldo, Pronóstico, Cuentas bancarias, Actualización conjunta, Tarjetas de crédito, Configurar, Generar a partir de plantilla, Archivos adjuntos, Impuestos, and Registration number search. Below the ribbon, the client's name '1765324498 : GUSTAVO LEORO R.' is displayed. The 'Datos demográficos de ventas' section is expanded, showing fields for 'Contacto principal', 'Línea de negocio', 'Empleado responsable' (Diego Gallo), 'Segmento', 'Subsegmento', and 'Cadena de empresas'. The 'Zona de ventas' field is highlighted with a red box and contains the value '02'. Other fields in this section include 'Moneda' (USD) and 'Notas'. Below this section are sections for 'Crédito y colecciones', 'Valores predeterminados de la orden de venta', and 'Valores predeterminados del pago'. The bottom status bar shows navigation icons and the text 'La zona de ventas del cliente'.

Fuente: Dynamics AX – Módulo de Gestión de Rutas

Para nuestro estudio se han predeterminado las siguientes zonas de venta dentro del Distrito Metropolitano de Quito, tomando como base las 8 administraciones zonales en las que se encuentra dividido.

*Tabla 3 Administraciones zonales (Zonas de Venta)*

Código	Descripción
01	La Delicia
02	Calderón

03	Quito Norte
04	Quito Centro
05	Quito Sur
06	Quitumbe
07	Valle de los Chillos
08	Valle de Cumbaya

Fuente: <http://www.quito.gob.ec/>

Esta clasificación permitirá estructurar cada una de las rutas en función de las zonas de ventas antes mencionadas, haciendo esta tarea mucho más sencilla.

Por otra parte, es imprescindible para nuestro estudio, definir la ubicación de la sucursal o sucursales de cada cliente.

El sistema ERP provee de forma estándar el ingreso de la siguiente información relacionada con la ubicación geográfica del cliente

*Gráfico 5 Información geográfica de clientes*

Fuente: Dynamics AX – Módulo de Gestión de Rutas

*Tabla 4 Información geográfica de clientes*

Campo	Descripción
País / región	Permite especificar el país de ubicación del cliente



Estado	Para nuestro propósito, corresponde a la provincia en la cual se ubica el cliente.
Ciudad	Corresponde a la ciudad en la cual se ubica el cliente
Distrito	Para nuestro caso de estudio corresponde a la parroquia en la que se ubica un determinado cliente.
Descripción	Una descripción general de la dirección del cliente
Calle	Ubicación del cliente

Fuente: Dynamics AX – Módulo de Gestión de Rutas

De igual forma cada dirección registrada y asociada a un cliente deberá poseer la información de latitud y longitud, información necesaria para poder desplegar en nuestra aplicación móvil, con la ayuda visual de un mapa, los puntos de visita programados para un día específico.

Gráfico 6 Coordenadas geográficas de Clientes

**General**

Impuestos sobre las ventas:

Zona horaria:

Latitud: -0,1580560000

Longitud: -78,4808330000

Vigente: 29/03/2015 15:11:05

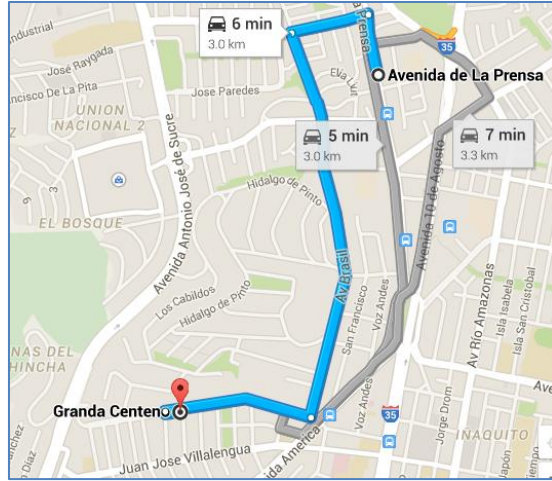
Caducidad: Nunca

Fuente: Dynamics AX – Módulo de Gestión de Rutas

### RUTA (PLANTILLA DE RUTAS)

Corresponde a un conjunto de clientes que deben ser visitados por el vendedor en un día específico de la semana. Como se había mencionado en la sección anterior, un cliente debe tener claramente definido la ubicación de sus sucursales tanto a nivel de dirección como de coordenadas de latitud y longitud.

*Gráfico 7 Rutas de Venta*



**Fuente: Google Maps**

De forma estándar no existe esta funcionalidad dentro del ERP, por lo tanto se ha desarrollado un módulo que permite gestionar y administrar la información de rutas.

Para la creación de una ruta de venta, se deben tener en cuenta los siguientes aspectos:

- **Identificador de Ruta:** Un número secuencial único que permite identificar a cada ruta.
- **Descripción:** Corresponde a una breve descripción de la ruta.
- **Zona de ventas:** Esta característica permite establecer rutas en función de las zonas de ventas propuestas.

Gráfico 8 Programación de Ruta de Venta

Fuente: Dynamics Ax – Módulo de Gestión de Rutas

- Vendedor: Permite identificar el vendedor asociado a una determinada ruta de venta y así conocer con claridad la planificación de visitas para cada día de la semana.
- Activo: Permite identificar si una ruta se encuentra activa. Las rutas que no se encuentren activas no serán consideradas en el proceso de envío de información hacia el dispositivo móvil.

Gráfico 9 Programación de Ruta de Venta

<b>Lunes</b>					
<div>  Nuevo            Eliminar         </div>					
<input type="checkbox"/>	Código de cliente	Nombre	Referencia de dirección	Descripción	Calle
	17437421001	IDEAL ALAMREC	5637158077	IDEAL ALAMREC	AV. LA PRENSA Y TELÉGRAFO I
	cli300001	Novacero	5637158078	direccion	AV. CORUÑA Y FRANCISCO DE ORELLANA
	1765324498	GUSTAVO LEORO R.	5637158079	SUCURSAL I	PEDRO DEL SOLAR E4-49 Y GUEPI
	1725097492001	RIO NAPO	5637158080	MATRIZ PRINCIPAL	URBANIZACIÓN JARDINES DE AMAGASI
<div> <div>▶ <b>Martes</b></div> <div>▶ <b>Miércoles</b></div> <div>▶ <b>Jueves</b></div> <div>▶ <b>Viernes</b></div> <div>▶ <b>Sábado</b></div> <div>▶ <b>Domingo</b></div> </div>					

Fuente: Dynamics Ax – Módulo de Gestión de Rutas

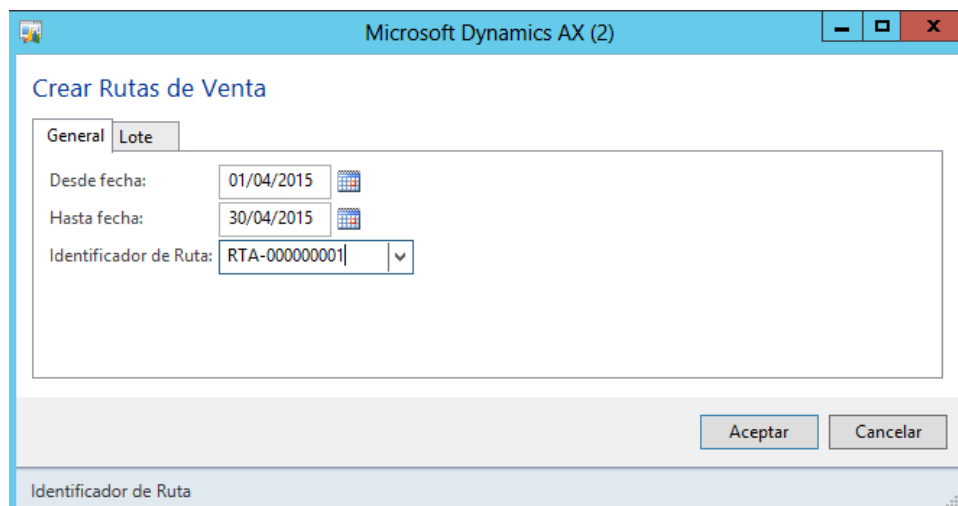
Clientes: Para cada día de la semana se asigna uno o varios clientes, desplegando únicamente aquellos clientes que se encuentren dentro de la zona de ventas configurada en la ruta. Adicionalmente, es importante especificar la dirección del cliente a visitar, ya que un cliente puede tener más de una dirección registrada.

Es importante destacar que esta es la plantilla predeterminada de configuración de rutas y sirve como base para la generación de rutas en un periodo de tiempo.

## GENERACIÓN DE RUTA

Esta sección permite generar, en un periodo específico de tiempo, todas las rutas o un grupo específico de rutas, en función de la información registrada en la sección de plantillas de ruta.

*Gráfico 10 Generación masiva de Rutas*



**Fuente: Dynamics Ax – Módulo de Gestión de Rutas**

Con esto, se obtiene un detalle diario las actividades de cada vendedor, con la posibilidad de hacer cambios en un día específico, ya sea agregar a una ruta específica la visita de un cliente diferente a la planificada, o en su defecto eliminar de una ruta la visita a un cliente.

La sesión mostrada en el gráfico No. 11, permite efectuar cualquier cambio requerido sobre una ruta específica en un día específico del año, de esta manera se consigue una total flexibilidad ante cambios inesperados en la programación de un vendedor.

Gráfico 11 Visualización detallada de rutas

Detalle de Rutas de Venta (2 - ac)

Archivo

Fecha	Día
01/03/2015	Domingo
02/03/2015	Lunes
03/03/2015	Martes
04/03/2015	Miércoles
05/03/2015	Jueves
06/03/2015	Viernes
07/03/2015	Sábado
08/03/2015	Domingo
09/03/2015	Lunes
10/03/2015	Martes
11/03/2015	Miércoles

+ Agregar - Quitar

Identificador de Ruta	Código de cliente	Nombre	Referencia de dirección	Calle	Vendedor	Nombre del trabajador
RTA-000000001	17437421001	IDEAL ALAMREC	5637158077	AV. LA PRENSA Y TELÉGRAFO I	gleoro	José Gustavo Leoro
RTA-000000001	cli300001	Novacero	5637158078	AV. CORUÑA Y FRANCISCO DE ...	gleoro	José Gustavo Leoro
RTA-000000001	1765324498	GUSTAVO LEORO R.	5637158079	PEDRO DEL SOLAR E4-49 Y GUEPI	gleoro	José Gustavo Leoro
RTA-000000001	1725097492001	RIO NAPO	5637158080	URBANIZACIÓN JARDINES DE A...	gleoro	José Gustavo Leoro

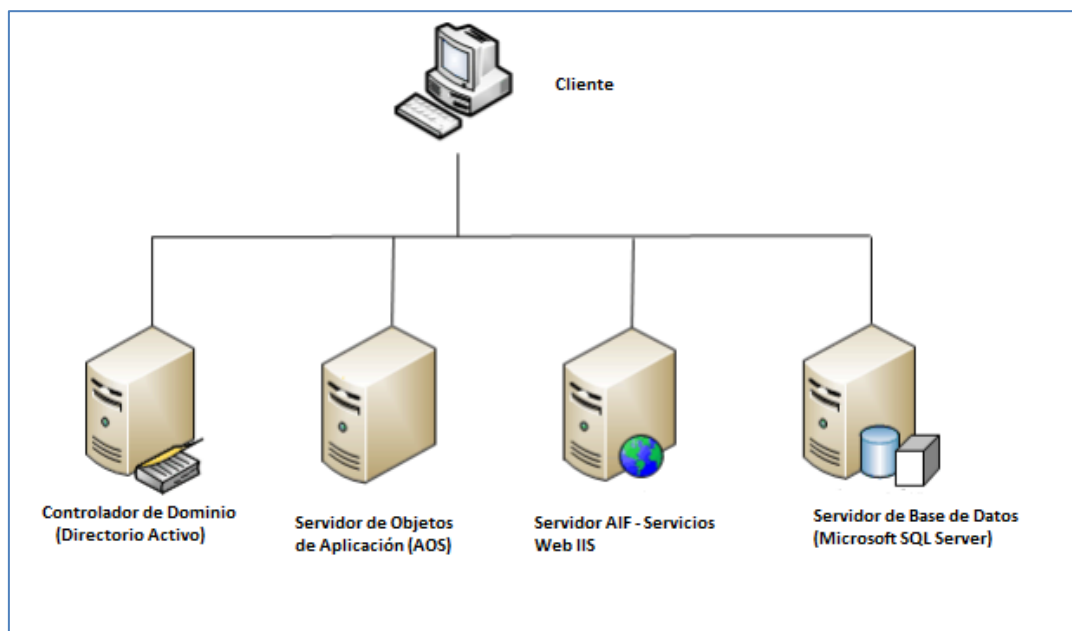
⏪ ⏴ ⏵ ⏩ 📄 📁 🔔(46) USD ac Cerrar

Fuente: Dynamics Ax – Módulo de Gestión de Rutas

### 5.3 Arquitectura e Infraestructura Microsoft Dynamics Ax

Es importante conocer los elementos básicos que conforman la configuración típica de una instalación de Microsoft Dynamics Ax.

Gráfico 12 Topología básica - Microsoft Dynamics Ax



Fuente: *Microsoft Dynamics Ax 2012 Installation Guide*

Controlador de Dominio: Los controladores de dominio tienen una serie de responsabilidades. Una de ellas es la autenticación. La autenticación es el proceso de garantizar o denegar a un usuario el acceso a recursos compartidos.

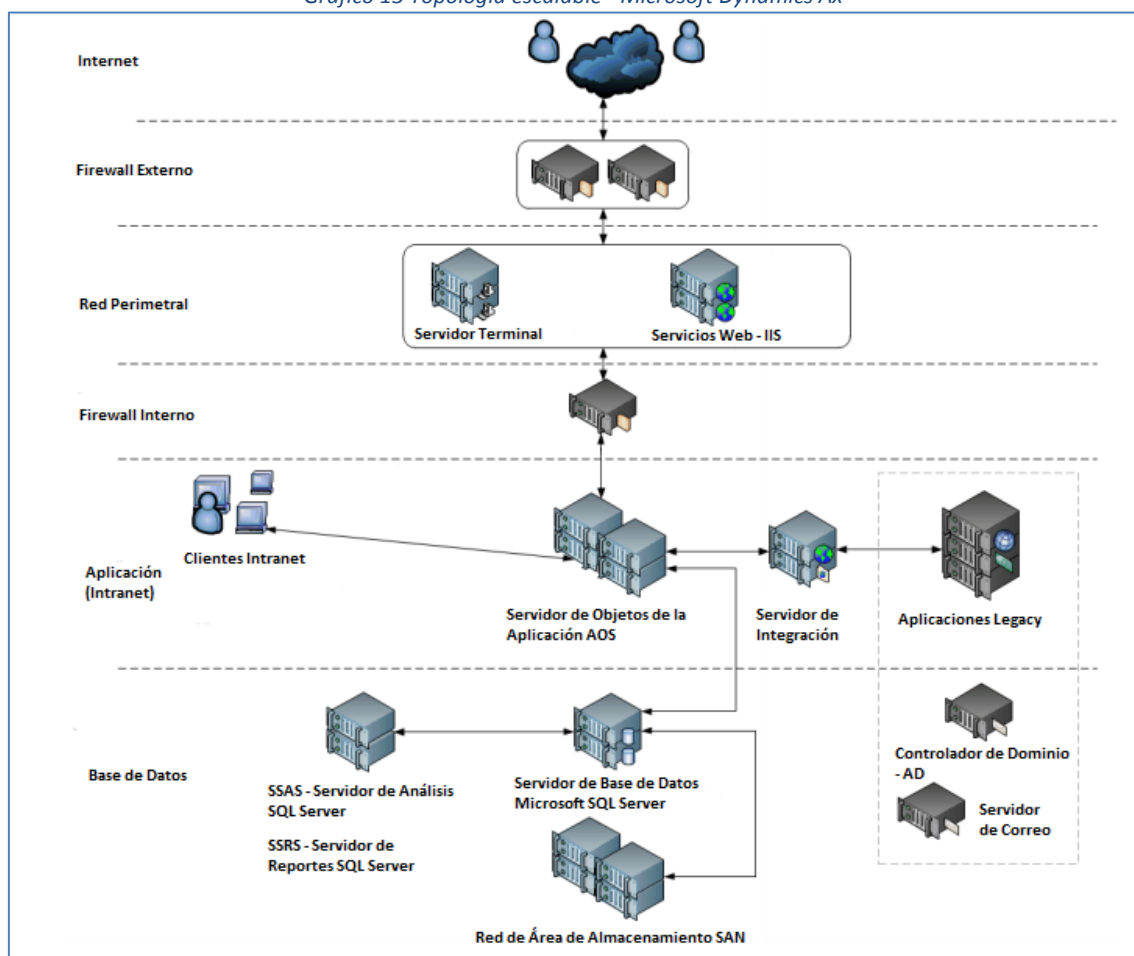
Servidor de Objetos de Aplicación (AOS): El servidor de objetos de la aplicación de Microsoft, por sus siglas en inglés AOS, viene a ser el componente central en una instalación de Microsoft Dynamics Ax, su función es robustecer la seguridad, administrar las conexiones entre un determinado cliente y el servidor de base de datos y proveer la base adecuada para ejecutar la lógica del negocio de Microsoft Dynamics Ax.

Servidor de Base de Datos: El servidor de base de datos aloja la base de datos que almacena toda la información transaccional generada en la aplicación. Este servidor también aloja una segunda base de datos que almacena todos los elementos de la aplicación, estos elementos incluyen cualquier personalización adicional realizada al ERP, como es el caso del módulo de administración de rutas propuesta en el presente trabajo de titulación.

Servidor AIF/WEB: El servidor del marco de trabajo de integración de aplicaciones por sus siglas en inglés AIF, aloja toda la funcionalidad expuesta por el ERP a través de servicios Web.

En la práctica, y dependiendo del presupuesto del departamento de IT de cada organización, el objetivo es conseguir una topología escalable y de alta disponibilidad.

Gráfico 13 Topología escalable - Microsoft Dynamics Ax



Fuente: Microsoft Dynamics Ax 2012 Installation Guide

El diagrama propuesto se basa en la arquitectura e infraestructura actual de la empresa Confiteca, este esquema muestra una topología estructurada en capas que proporciona mayor seguridad mediante la ubicación estratégica de cortafuegos y el uso de una red perimetral o zona desmilitarizada por sus siglas en inglés DMZ. Una red perimetral, evita el acceso directo de la intranet por parte de cualquier usuario externo.

Dentro del intranet corporativo, los servidores están divididos en función de las siguientes capas:

- *Capa de aplicación (intranet)*, que contiene aquellos servidores que se especializan en proporcionar información a los clientes, cómo es el caso del AOS.
- *Capa de base de datos*, que contiene aquellos servidores especializados en el almacenamiento de información y que pueden ser recuperados por los servidores localizados en la capa de aplicación. En esta capa también se ubican servidores que proporcionan a toda la empresa funciones administrativas y de seguridad, como es el caso del servidor de directorio activo y de correo electrónico.

Los clientes de Microsoft Dynamics Ax que se conectan a través de una red WAN están configurados para utilizar los servicios de terminal server para comunicarse con el servidor de objetos de aplicación (AOS). Los clientes de Microsoft Dynamics Ax que se conectan a través de una red LAN (intranet) están configurados para conectarse con el servidor de objetos AOS directamente.

### Red Perimetral

La red perimetral proporciona acceso a los usuarios externos a la funcionalidad de Microsoft Dynamics Ax a través de los siguientes tipos de servidores:

- *Servidor de Terminal*, que proporciona, a través de una red privada virtual, acceso a usuarios autorizados.
- Para nuestra llevar a cabo nuestra propuesta, se incluye en la infraestructura actual de la organización un servidor web (IIS) que proporciona acceso a la funcionalidad de Microsoft Dynamics Ax mediante el uso de servicios web.



### Capa de Aplicación

La capa de aplicación contiene servidores que proporcionan información tanto a clientes internos como externos que acceden a Microsoft Dynamics Ax a través de la red perimetral. Esta capa contiene los siguientes servidores:

- Servidor AOS: El servidor AOS proporciona comunicación entre los clientes de Ax y el servidor de base de datos. Adicionalmente gestiona funcionalidad para la integración con sistemas externos (AIF), tareas de procesamiento en lote y actividades de flujo de trabajo.
- Servidor de Integración: Proporciona conectividad con sistemas pre-existentes.

### Capa de Base de Datos

La capa de base de datos contiene los siguientes servidores:

- Servidor Microsoft SQL Server: contiene la base de datos de la aplicación Microsoft Dynamics Ax
- Servidor SSAS: Servidor de análisis de datos que gestiona y administra la rama correspondiente a la inteligencia de negocios (Cubos de Información)
- Servidor SSRS: Servidor de Informes que gestiona la implementación y administración de los informes generados desde la aplicación de Microsoft Dynamics Ax.

En cuanto a los requerimientos de red, El ERP puede operar sobre redes que ejecuten el protocolo IPv4 o el protocolo IPv6. LA velocidad mínima de conexión requerida entre el servidor AOS y el servidor de base de datos es de 100 Mbps (b).

El tiempo de respuesta de ejecución de una tarea sobre el ERP viene expresada por la siguiente fórmula:

$$\text{Tiempo de Respuesta} = (\text{Número de llamadas hacia y desde el servidor}) * \text{Latencia} + \text{Número de bytes enviados} / b$$

De la fórmula se puede concluir que mientras mayor es la latencia de la red, mayor será el tiempo de respuesta.

## 5.4 Esquema de Integración

El ERP proporciona un marco de trabajo que permite la integración con aplicaciones externas. Este marco de trabajo se conoce como marco de integración de aplicaciones por sus siglas en inglés AIF.

El marco de integración de aplicaciones (AIF) habilita a la organización la posibilidad de comunicar e integrarse con procesos externos o socios estratégicos mediante el intercambio de documentos XML a través de diversos medios y adaptadores de transporte, el presente trabajo de titulación se centra en el adaptador HTTP que permite la comunicación sincrónica con un sitio web.

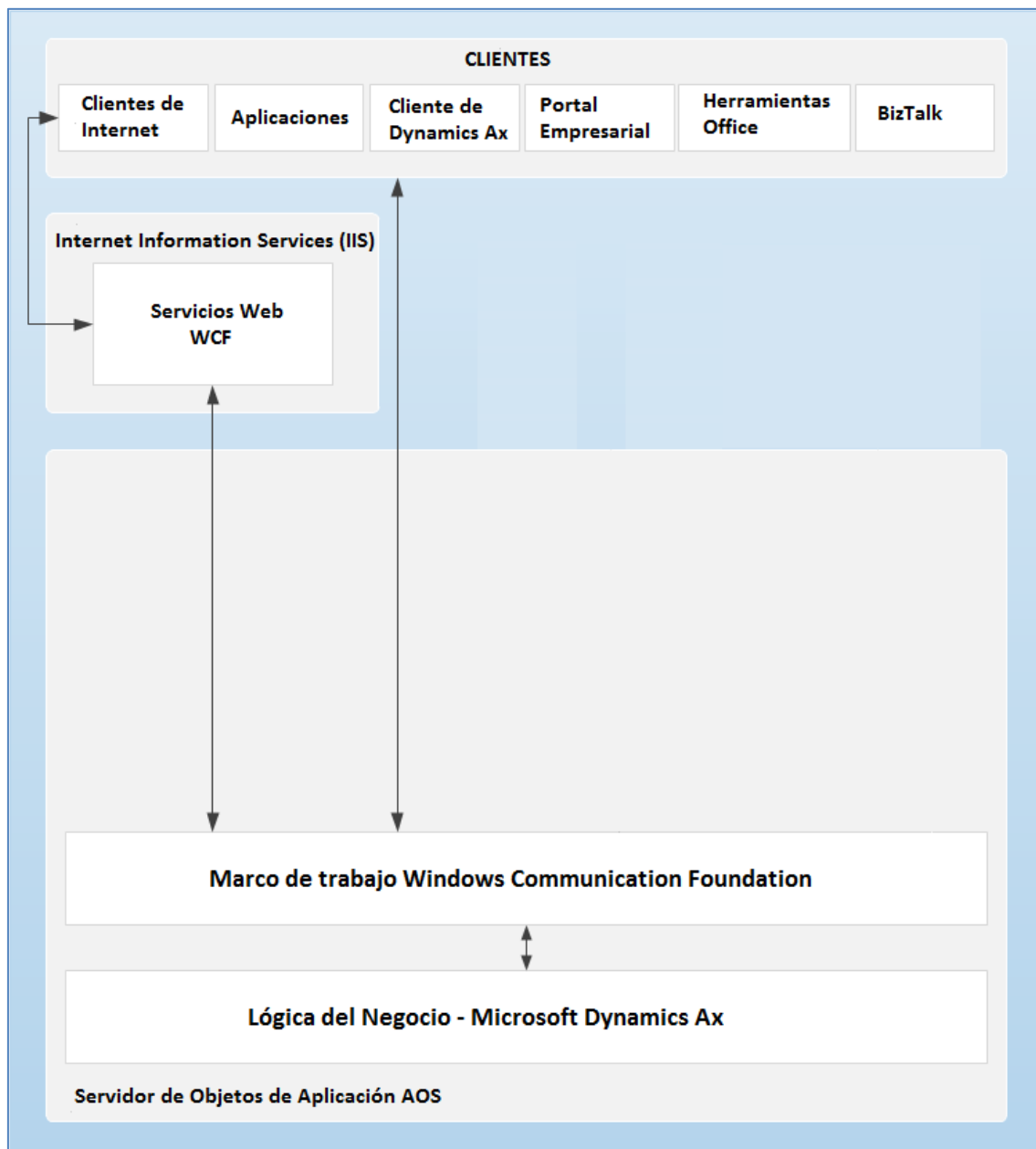
El ERP expone su funcionalidad a través de servicios que están basados en Windows Communication Foundation (WCF) y que se encuentran alojados en el servidor de objetos de la aplicación AOS.

Tanto las aplicaciones externas como aplicaciones y componentes propias del ERP, consumen los servicios de Microsoft Dynamics Ax accediendo directamente hacia el servidor AOS.

Aplicaciones externas basadas en internet, como la aplicación propuesta en este trabajo de titulación, acceden a los servicios de Microsoft Dynamics Ax a través del servidor de Internet Information Services (IIS). IIS enruta todas las peticiones entrantes de los servicios del ERP hacia el servidor de objetos de aplicación (AOS).

Todas las solicitudes de petición, independientemente de su origen, son gestionadas en tiempo de ejecución por WCF que se encuentra alojado en el servidor de objetos de la aplicación AOS.

Gráfico 14 Arquitectura del Marco de Integración de Aplicación - Microsoft Dynamics Ax



Fuente: *Microsoft Dynamics Ax 2012 Installation Guide*

## 5.5 Consideraciones a nivel de seguridad

La exposición de los servicios web en el ERP, se lo realiza a través de los denominados puertos de integración que permiten administrar y exponer uno varios servicios web. Cada puerto de integración posee un único identificador de recurso (URI) que identifica la dirección del puerto. Por defecto todos los usuarios definidos en el ERP (sección estándar de administración de usuarios del ERP) tienen acceso a los puertos de integración, sin embargo es recomendable restringir su acceso a grupos o usuarios específicos.

El segundo nivel de seguridad aplicado en el trabajo de titulación corresponde a la configuración de usuarios intermediarios de confianza, este tipo de usuario puede enviar solicitudes de petición a un puerto de integración en nombre de otro usuario; este usuario debe pertenecer al directorio activo de la empresa, tener acceso al puerto de integración donde se expone el servicio web y finalmente ser un usuario válido dentro del ERP.

El o los usuarios a los que suplanta el usuario intermediario de confianza, corresponde a usuarios que no necesariamente pertenecen al directorio activo (para el presente trabajo de titulación, estos usuarios corresponden a los vendedores) pero que obligatoriamente deben estar registrados en la ficha de administración de usuarios del ERP con el tipo “usuario de reclamación” (claim user); de esta forma se garantiza y se delega la autenticación al propio ERP.

Es por ello que en cada invocación a un servicio web, la aplicación móvil envía siempre el identificador de usuario.

En resumen, la seguridad en la exposición y consumo de los servicios web se aplica a través de la combinación de WCF, IIS, el directorio activo y la seguridad propia del ERP a través del manejo de roles de seguridad.

- Cualquier petición o invocación a un servicio web es recibida por el servidor web (IIS) donde estos servicios se encuentran alojados. IIS recupera las credenciales del usuario que realiza la petición y lo valida contra el listado de usuarios registrados en el directorio activo.

- A continuación, la solicitud es procesada por el servidor AIF el cual ejecuta una autenticación adicional verificando que el usuario solicitante cumpla con los siguientes criterios:
  - Sea un usuario válido del ERP
  - Tengo los roles y permisos necesarios dentro del ERP

## 5.6 Servicios Web personalizados en Microsoft Dynamics Ax

El ERP posee una gama de servicios predeterminados para efectuar muchas de las tareas comunes dentro del sistema central, sin embargo al tratarse de un módulo completamente nuevo, es necesario recurrir a la creación de servicios personalizados que permitan exponer cualquier nueva funcionalidad desarrollada bajo X++<sup>1</sup> hacia cualquier tipo de cliente externo.

Cualquier funcionalidad desarrollada bajo x++ puede ser expuesta a través de un servicio web personalizado mediante el uso de contratos de datos (data contracts).

Un contrato de datos, es un acuerdo formal entre un servicio y un cliente donde se describe los datos que se intercambiarán (datos miembro). Un contrato de datos define con precisión, para cada parámetro o tipo de valor devuelto, que datos se serializan<sup>2</sup> para el intercambio de información.

Para entender mejor utilizaremos como ejemplo el servicio personalizado que permite el intercambio de información del stock de producto.

Para el intercambio de información del stock de producto se ha definido el siguiente contrato de datos (clase). Esta clase denominada **GLDetalleInventarioVendedorContract** especifica todos los atributos (datos miembros) que estarán involucrados en el intercambio de información. Para identificar a un contrato de datos y a un dato miembro se utiliza la palabra reservada `DataContractAttribute` y `DataMemberAttribute` respectivamente.

---

<sup>1</sup> Lenguaje de programación orientado a objetos empleado para agregar o modificar funcionalidades en Microsoft Dynamics Ax

<sup>2</sup> Consiste en un proceso de codificación de un objeto con el fin de transmitirlo a través de una conexión en red como una serie de bytes o en un formato humanamente más legible como XML o JSON

```

[DataContractAttribute]
class GLDetalleInventarioVendedorContract
{
    ItemId codigoProducto;
    ItemName nombreProducto;
    UnitOfMeasureSymbol unidadProducto;
    Price precioProducto;
    InventQtyAvailPhysical stockProducto;
}

```

Este contrato se compone de los siguientes datos miembros:

- Código de Producto:

```

[DataMemberAttribute('codigoProducto')]
public ItemId codigoProducto(ItemId _codigoProducto = codigoProducto)
{
    codigoProducto = _codigoProducto;
    return codigoProducto;
}

```

- Nombre de Producto:

```

[DataMemberAttribute('nombreProducto')]
public ItemName nombreProducto(ItemName _nombreProducto = nombreProducto)
{
    nombreProducto = _nombreProducto;
    return nombreProducto;
}

```

- Precio del Producto:

```

[DataMemberAttribute('precioProducto')]
public Price precioProducto(Price _precioProducto = precioProducto)
{
    precioProducto = _precioProducto;
    return precioProducto;
}

```

- Inventario disponible:

```

[DataMemberAttribute('stockProducto')]
public InventQtyAvailPhysical stockProducto(InventQtyAvailPhysical _stockProducto
=stockProducto)
{
    stockProducto = _stockProducto;
    return stockProducto;
}

```

- Unidad de Venta del producto:

```

[DataMemberAttribute('unidadProducto')]
public UnitOfMeasureSymbol unidadProducto(UnitOfMeasureSymbol _unidadProducto =
unidadProducto)
{
    unidadProducto = _unidadProducto;
    return unidadProducto;
}

```

El siguiente fragmento de código, se trata de un método que recibe como parámetro el código del vendedor, y en función del almacén atado al vendedor retorna un listado de todos los productos disponibles en ese almacén.

```
[SysEntryPointAttribute(true), AifCollectionTypeAttribute('return', Types::Class,
classStr(GLDetalleInventarioVendedorContract))]  
  
public List obtenerInformacionInventario(HcmPersonnelNumberId _vendedor)  
{  
    List          resultSet = new List(Types::Class);  
    GLVendedorMovil vendMovil;  
    InventSum isum;  
    InventDim idim;  
    InventTable itable;  
    InventTableModule imodule;  
    GLDetalleInventarioVendedorContract inventario;  
    ;  
    while select vendMovil  
    index hint idx  
    where vendMovil.Empleado==_vendedor  
    join ReservPhysical,PostedQty,Received,Deducted,Registered,Picked, ItemId from isum  
    join InventLocationId, InventSiteId from idim  
    where idim.inventDimId==isum.InventDimId  
    && idim.InventLocationId==vendMovil.Almacen  
    && idim.InventSiteId==vendMovil.Sitio  
    join Price,UnitId from imodule  
    index hint ItemModuleIdx  
    where imodule.ModuleType==ModuleInventPurchSales::Sales  
    && imodule.ItemId==isum.ItemId  
    {  
        inventario = new GLDetalleInventarioVendedorContract();  
        inventario.codigoProducto(isum.ItemId);  
        inventario.nombreProducto(InventTable::find(isum.ItemId).itemName());  
        inventario.precioProducto(imodule.Price);  
        inventario.unidadProducto(imodule.UnitId);  
        inventario.stockProducto(isum.availPhysicalCalculated());  
        if(isum.availPhysicalCalculated()>0)  
            resultSet.addEnd(inventario);  
    }  
  
    return resultSet;  
}
```

Para indicar que se trata de una funcionalidad que será expuesta a través de un servicio web se utiliza el atributo SysEntryPointAttribute configurado con el valor True.

Debido a que esta operación de servicio retorna una lista que contiene en su interior uno o varios objetos de la clase GLDetalleInventarioVendedorContract con la información detallado de cada producto y que son propios del lenguaje x++, es necesario hacer uso del atributo AifCollectionTypeAttribute para especificar el tipo de dato que retornará.

## 5.7 Esquema de intercambio de información

En la sección 5.4 se ha mencionado mucho el término Windows Communication Foundation, por lo tanto es importante explicar el significado de este concepto.

Windows Communication Foundation (WCF) es un marco de trabajo para la creación de aplicaciones orientadas a servicios. Al hablar de arquitectura orientada a servicios, nos referimos al uso de servicios web para enviar y recibir datos.

Windows Communication Foundation permite enviar datos como mensajes asincrónicos de un extremo de servicio a otro. Un extremo está compuesto por una ubicación (dirección URL) que define a donde se pueden enviar los mensajes, un enlace que describe como se debe enviar el mensaje y un contrato de servicio que describe que mensajes se pueden enviar.

Otra de las ventajas a destacar y que es clave en el desarrollo del trabajo de titulación propuesto, es que WCF es totalmente compatible con la tecnología REST y admite formatos no XML como la notación de objetos de Javascript por sus siglas en inglés JSON, que corresponde a la estructura utilizada en el intercambio de mensajes para la aplicación propuesta.

### SERVICIOS RESTFUL

Proviene del término REST que significa Transferencia de Estado Representacional y se trata de una arquitectura de software utilizada para el diseño de servicios web orientada a recursos. Las arquitecturas tradicionales orientadas a servicios como SOAP usan el concepto de llamadas a procedimientos remotos (RPC), es decir, a invocar métodos sobre un servicio remoto. Con Rest se opera sobre recursos y no sobre servicios.

Entre las características principales, podemos mencionar:

- Utiliza los métodos HTTP de manera explícita
- Se accede a cada recurso mediante URI
- Transfiere información en formato XML y JSON



Una de las características claves de los servicios web REST es el uso explícito de los métodos HTTP de acuerdo al protocolo definido por RFC 2616<sup>3</sup>. REST hace uso de los siguientes métodos para cada una de las actividades que se menciona a continuación:

POST: Se utiliza para crear un recurso en el servidor

GET: Para obtener un recurso en el servidor

PUT: Para cambiar el estado de un recurso o actualizarlo

DELETE: Para eliminar un recurso.

Por otra parte, REST expone cada uno de los recursos a través de un identificador uniforme de recursos (URI).

Para el trabajo de titulación propuesto, la interacción entre la aplicación móvil y Microsoft Dynamics Ax se lo realiza a través de los siguientes recursos:

- Validación de Usuarios:
  - <http://181.39.31.3/WSAxMovil/AXMovil.svc/autenticarUsuario>
- Carga de Rutas:
  - <http://181.39.31.3/WSAxMovil/AXMovil.svc/cargarInformacionRuta>
- Retorno de Pedido:
  - <http://181.39.31.3/WSAxMovil/AXMovil.svc/crearPedido>
- Listado de Vendedores
  - <http://181.39.31.3/WSAxMovil/AXMovil.svc/cargarListadoVendedores>
- Ventas efectivas por vendedor
  - <http://181.39.31.3/WSAxMovil/AXMovil.svc/cargarInformacionVentas>
- Listado de Clientes
  - <http://181.39.31.3/WSAxMovil/AXMovil.svc/cargarListadoClientes>
- Listado de Sucursales por Cliente

---

<sup>3</sup> <https://www.ietf.org/rfc/rfc2616.txt>

- <http://181.39.31.3/WSAxMovil/AXMovil.svc/cargarListadoDirecciones>
- Ingreso de Sucursal
  - <http://181.39.31.3/WSAxMovil/AXMovil.svc/agregarSucursal>
- Edición de Sucursal
  - <http://181.39.31.3/WSAxMovil/AXMovil.svc/editarSucursal>

Los formatos de cada uno de los parámetros de entrada así como el formato de salida de cada uno de estos servicios web se explican en la sección 5.7.

## JSON

JSON, es un formato ligero de intercambio de datos, que se caracteriza por su simplicidad en la lectura y escritura para un humano y en su generación e interpretación por diversos lenguajes de programación. Está basado en un subconjunto del lenguaje de programación de JavaScript y su fortaleza radica en su independencia de un lenguaje de programación específico haciéndolo ideal para el intercambio de datos.

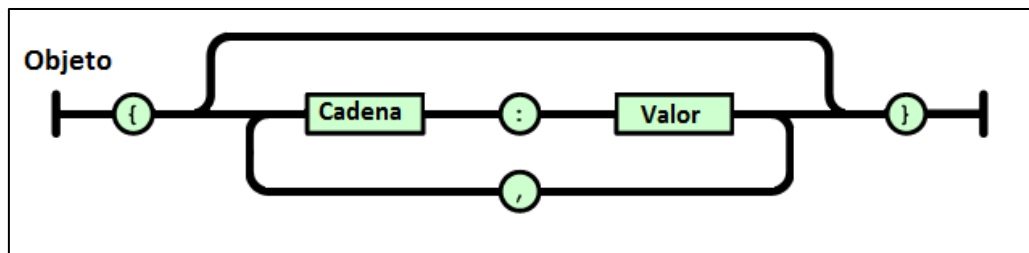
JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores o listas.

En JSON, se presentan de estas formas:

- Un *objeto* es un conjunto desordenado de pares nombre/valor. Un objeto comienza con { (llave de apertura) y termina con } (llave de cierre). Cada nombre es seguido por : (dos puntos) y los pares nombre/valor están separados por , (coma).

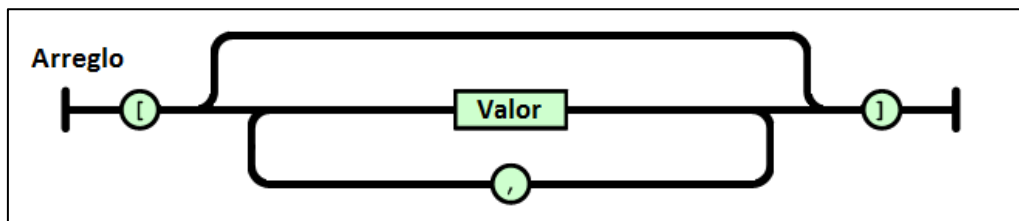
Gráfico 15 Componentes de la notación JSON



Fuente: <http://json.org/json-es.html>

- Un arreglo, constituye una colección de valores. Un arreglo comienza con [ (corchete de apertura) y termina con ] (corchete de cierre). Los valores se separan por , (coma)

Gráfico 16.- Componentes de la notación JSON



Fuente: <http://json.org/json-es.html>

- Un valor constituye una cadena de caracteres con comillas dobles, o un número, o valores booleanos, valores nulos, objetos o arreglos.

Gráfico 17 Componentes de la notación JSON



Fuente: <http://json.org/json-es.html>

## 5.8 Estructura de documento de Servicios

Para la gestión de intercambio de información entre el sistema ERP Microsoft Dynamics Ax y el dispositivo móvil, se ha definido los siguientes servicios web, cuyo formato de entrada y salida de datos se detalla a continuación.

### VALIDACIÓN DE USUARIO

Cómo ya se había mencionado anteriormente la gestión y administración de usuarios se lo realiza desde el sistema central ERP, por ello la necesidad de generar el primer servicio web para el intercambio de información.

El servicio web de validación de usuarios tiene la siguiente estructura de entrada y salida de datos.

#### Entrada:

```
{  
    "userid": "gleoro",  
    "password": "12345678"  
}
```

#### Salida:

```
{  
    "access": true/false,  
    "userid": "gleoro"  
    "role": "seller/administrator"  
}
```

Este servicio web recibe como datos de entrada dos datos de tipo alfanumérico que corresponden al nombre de usuario (código de empleado en el sistema ERP) y la contraseña proporcionada por el administrador del módulo de rutas de venta en Microsoft Dynamics Ax.

Si las credenciales han sido validadas correctamente, el servicio web retorna en la cadena "**Access**" el valor true, en la cadena "**userid**" el nombre de usuario y en la cadena "role" el perfil del usuario; caso contrario retorna en la cadena "**Access**" el valor false y en la cadena "**userid**" y "**role**" el valor nulo.

## ENVIO DE INFORMACIÓN DE RUTAS Y STOCK DE PRODUCTOS

El módulo de administración de rutas planteado para Microsoft Dynamics Ax, permite asignar a cada vendedor en una fecha específica una agenda de visitas a determinados clientes, esta agenda de visitas se conoce como ruta de ventas el cual es gestionado íntegramente desde el sistema central ERP.

Para cumplir con el envío de esta información, se ha generado un servicio web que cumple con la siguiente estructura de entrada y salida de datos.

### **Entrada:**

```
{
  "userid": "gleoro",
  "date": 1425859200
}
```

### **Salida:**

```
{
  "userid": "gleoro",
  "date": 1425859200,
  "route": [
    {
      "latitude": -0.198611, "longitude": -78.482222, "customerid": "001",
      "branchOfficeld": 5637158077, "customerName": "Nombre Cliente",
      "address": "Dirección Cliente", "customerDocld": 1715960173001,
      "socialName": "Razón Social"
    }
  ],
  "inventory": [
    {
      "productld": "001",
      "productName": "Caramelos", "unitld": "Kg.", "stock": 500, "price": 10
    }
  ]
}
```

Este servicio web recibe como datos de entrada el nombre de usuario, que corresponde al código del vendedor dentro de Microsoft Dynamics Ax y la fecha de solicitud que se encuentra en notación tipo UNIX y que representa el número de segundos transcurridos desde el 1 de enero de 1970.

Si dentro de la planificación de rutas, existe una ruta activa para el vendedor y la fecha de solicitud, el servicio web retorna la siguiente información que se compone de dos grandes estructuras:

- Rutas (route): Está compuesto por uno o más clientes que están dentro de la planificación de visitas de un determinado vendedor para un día específico. Internamente contiene los siguientes valores:
  - **Latitude:** Corresponde a las coordenadas de latitud de la dirección del cliente
  - **Longitud:** Corresponde a las coordenadas de longitud de la dirección del cliente
  - **CustomerId:** Corresponde al código único del cliente con el cual se lo identifica en el sistema Microsoft Dynamics Ax
  - **CustomerName:** Corresponde al nombre del cliente a visitar.
  - **CustomerDocId:** Corresponde al número de identificación del cliente (Cédula / RUC)
  - **SocialName:** Corresponde a la razón social del cliente a visitar.
  - **Address:** Corresponde a la descripción de la dirección del cliente a visitar.
  - **BranchOfficeld:** Corresponde al código identificador de la dirección del cliente
- Inventario o stock de producto (inventory): Como se había mencionado anteriormente un vendedor necesariamente debe estar atado a una bodega de inventario; bajo esta premisa, se envía hacia el dispositivo móvil todo el stock de producto disponible en esa bodega con el objetivo de que el vendedor tenga toda la información necesaria para generar los diversos pedidos de venta. Internamente contiene los siguientes valores:
  - **ProductId:** Corresponde al código único del cliente con el cual se lo identifica en el sistema Microsoft Dynamics Ax
  - **ProductName:** Corresponde al nombre del producto

- **UnitId**: Especifica la unidad de venta que maneja el producto.
- **Stock**: Representa la cantidad disponible de ese producto en la bodega de inventario atada al vendedor.

### RETORNO DE PEDIDO DE VENTA

Una vez generado el pedido de venta desde la aplicación móvil, el vendedor debe retornar este pedido de venta hacia el ERP para que sea procesado y registrado. Esta acción se lo realiza a través de un servicio web que tiene la siguiente estructura:

#### **Entrada:**

```
{
  "userId": "gleoro",
  "date": "1425859200",
  "customerId": "001",
  "latitude": "-0.198611",
  "longitude": "-78.482222",
  "branchOfficeId": "5637158077",

  "SalesOrderDetail":
    [
      {"productId": "001", "quantity": 10},
    ]
}
```

#### **Salida:**

```
{
  "successful": true/false
  "salesId": "OV-0001"
  "invoiceId": "FA00100200000001"
  "message": "Pedido creado exitosamente"
}
```

Este servicio web recibe como parámetro de entrada una estructura típica de un pedido de venta (orden de venta) que está conformado por una cabecera y un detalle el cual a su vez consta de varias líneas dependiendo del producto o productos seleccionados.

- **Userld:** Contiene el código del vendedor quien realiza el pedido de venta
- **Date:** Contiene la fecha en la cual se está generando la orden de venta. La fecha se encuentra en formato de tiempo Unix
- **Customerld:** Contiene el código del cliente al cual se está generando y registrando la factura de venta
- **Latitude:** Contiene las coordenadas de latitud desde el sitio en el cual se está registrando la orden de venta. Esta información la obtiene a partir del GPS del dispositivo móvil.
- **Longitude:** Contiene las coordenadas de longitud desde el sitio en el cual se está registrando la orden de venta. Esta información la obtiene a partir del GPS del dispositivo móvil.
- **BranchOfficeld:** Corresponde al código identificador de la dirección o sucursal del cliente donde se está realizando la entrega del producto.
- **SalesOrderDetail:** Lo mencionado en los puntos anteriores corresponde a la información de la cabecera de la orden de venta. A continuación se detalla los elementos necesarios requeridos en el detalle de orden de venta de acuerdo con Microsoft Dynamics Ax.
  - **Productld:** Corresponde al código único del cliente con el cual se lo identifica en el sistema Microsoft Dynamics Ax
  - **Quantity:** Corresponde a la cantidad despachada de cada producto.

Una vez procesada la información en Microsoft Dynamics Ax, el servicio web retorna la siguiente información:



- **Successful:** Si la orden de venta fue creada y registrada correctamente en el sistema ERP, el valor de esta cadena es "true". Si la orden de venta presentó algún problema durante el registro en Microsoft Dynamics Ax, el valor de esta cadena es "false".
- **SalesId:** Contiene el código de orden de venta generado en el sistema Microsoft Dynamics Ax.
- **InvoiceId:** Contiene el número de factura generado en el sistema Microsoft Dynamics Ax.
- **Message:** Contiene un mensaje informativo de éxito o de error que se mostrará en la aplicación móvil.

Para la sección de administración de la aplicación móvil, se han desarrollado los siguientes servicios web:

### ENVÍO DE LISTADO DE VENDEDORES

El administrador o los administradores de la aplicación podrán tener un seguimiento de las ventas efectivas realizadas por cada vendedor en un día específico.

El servicio web detallado a continuación permite obtener el listado de vendedores en función de las zonas de ventas anteriormente detalladas.

#### **Entrada:**

```
{
  "userid": "usuario"
  "zoneId": "zona de ventas"
}
```

#### **Salida:**

```
{
  "vendorList ":
  [
    {"name": "Gustavo Leoro", "vendorId": "gleoro"},
  ]
}
```

Este servicio web recibe como parámetro de entrada el código de la zona de ventas, en función de la tabla 3 especificado en la sección cliente del módulo de administración de rutas – Microsoft Dynamics Ax.

Como información de salida, el servicio web retorna un listado de vendedores que responde a la siguiente estructura:

- **VendorList:** Se compone de un listado de vendedores. La información proporcionada de cada vendedor se detalla a continuación:
  - **Name:** Nombre del vendedor, que corresponde al nombre del empleado dentro del módulo de recursos humanos de Microsoft Dynamics Ax
  - **VendorId:** Contiene el código del vendedor.

### ENVÍO DE VENTAS POR VENDEDOR

Este servicio web permite conocer, dado un vendedor y una fecha específica, las ventas efectivas realizadas en la ruta correspondiente a ese día y el total facturado para cada cliente.

Como parámetros de entrada, el servicio web recibe el código del vendedor (**VendorId**) que corresponde al código del empleado dentro del sistema ERP, y la fecha de solicitud bajo la notación de tiempo Unix (**Date**).

#### **Entrada:**

```
{  
  "userId": "usuario"  
  "vendorId": "gleoro",  
  "date": 1425859200  
}
```

#### **Salida:**

```
{  
  "vendorId": "gleoro",  
}
```

```

    "date": 1425859200
    "route":
    [
      {
        "address": "descripción direccion",
        "customerId": "código cliente",
        "customerName": "nombre cliente", "invoiceAmount": 0.00,
        "latitude": -0.158056,
        "latitudeSale": -0.158056, "longitude": -78.480833,
        "longitudeSale": -78.480833
      }
    ]
  }

```

La información retornada por el servicio web corresponde a la información de las ventas facturadas a cada cliente el cual se almacena en el arreglo **"route"** y contiene los siguientes objetos:

- **Route:** Almacena la información de las ventas efectuadas a cada cliente para una ruta específica
  - **Address:** Contiene la descripción de la dirección (sucursal) del cliente
  - **CustomerId:** Contiene el código del cliente a quien factura la orden de venta.
  - **CustomerName:** Contiene el nombre del cliente a quien se factura la orden de venta.
  - **InvoiceAmount:** Contiene el total de la factura registrada al cliente
  - **Latitude:** Contiene las coordenadas de latitud de la dirección de la sucursal del cliente
  - **LatitudeSale:** Contiene las coordenadas de latitud del sitio donde se registró la orden de venta.
  - **Longitude:** Contiene las coordenadas de longitud de la dirección de la sucursal del cliente
  - **LongitudeSale:** Contiene las coordenadas de longitud del sitio donde se registró la orden de venta.

## ENVÍO DE LISTADO DE CLIENTES

Este servicio web permite conocer información básica sobre los clientes almacenados en el sistema central ERP.

No recibe ningún parámetro de entrada, y como información de salida retorna el listado de clientes en base a la siguiente estructura:

**Salida:**

```
{
  "customerList":
  [
    {"customerDocId":"1718042672001","customerId":"0503",
    "customerName":"Nombre","socialName":"Razón Social"}
  ]
}
```

- **CustomerList:** Se compone de un listado de clientes. La información proporcionada de cada cliente se detalla a continuación:
  - **customerName:** Nombre del cliente, que corresponde al nombre del cliente dentro del módulo de cuentas por cobrar de Microsoft Dynamics Ax
  - **customerId:** Contiene el código del cliente.
  - **customerDocId:** Contiene el número de identificación del cliente
  - **SocialName:** Contiene la razón social del cliente.

## ENVÍO DE DIRECCIONES (SUCURSALES) DE CLIENTES

Para que el administrador de la aplicación móvil tenga la capacidad de registrar o actualizar la información geográfica de un cliente específico, se ha desarrollado este servicio web que cumple con la siguiente estructura.

Cómo parámetro de entrada, el servicio web recibe el código del cliente (**customerId**) que corresponde al código del cliente dentro del sistema ERP.

Entrada:

```
{
```

```
  "customerId": "0001"
```

```
}
```

Salida:

```
{
```

```
  "addressList":
```

```
    [{"address": "Av. La Prensa",
```

```
      "branchOfficeId": 5637160326, "city": "UIO",
```

```
      "description": "dirección de entrega",
```

```
      "district": "04", "latitude": 0, "longitude": 0, "state": "17"}]
```

```
  ],
```

```
  "customerDocId": "1718042672001",
```

```
  "customerId": "0001",
```

```
  "customerName": "GLR",
```

```
  "socialName": "GLR"
```

```
}
```

La información retornada por el servicio web corresponde a cada una de las sucursales (direcciones) registradas en el sistema ERP y que responden a la siguiente estructura:

- **customerName:** Nombre del cliente, que corresponde al nombre del cliente dentro del módulo de cuentas por cobrar de Microsoft Dynamics Ax
- **customerId:** Contiene el código del cliente.
- **customerDocId:** Contiene el número de identificación del cliente
- **SocialName:** Contiene la razón social del cliente.
- **AddressList:** Arreglo que almacena la información de las sucursales de cada cliente.
  - **Address:** Contiene la descripción de la dirección (sucursal) del cliente
  - **Latitude:** Contiene las coordenadas de latitud de la dirección de la sucursal del cliente
  - **Longitude:** Contiene las coordenadas de longitud de la dirección de la sucursal del cliente
  - **BranchOfficeId:** Corresponde al código identificador de la dirección o sucursal del cliente.

- **Description:** Contiene una descripción adicional de la dirección (sucursal) del cliente
- **City:** Contiene el código de la ciudad donde está ubicada la sucursal del cliente
- **State:** Contiene el código de la provincia donde está ubicada la sucursal del cliente
- **District:** Contiene el código de la parroquia donde está ubicada la sucursal del cliente

### ACTUALIZACIÓN DE DIRECCIONES (SUCURSALES) DE CLIENTES

Otra de las características que brinda la aplicación, es facilitar la actualización de la ubicación del cliente a través del uso del dispositivo móvil y su GPS.

Para que el administrador de la aplicación móvil tenga la capacidad de actualizar la información geográfica de un cliente específico, se ha desarrollado este servicio web que cumple con la siguiente estructura.

**Entrada:**

```
{
    "branchOfficeld":55455453,
    "latitude": -0.7736362,
    "longitude": -17.545443,
    "state": "17",
    "city": "UIO",
    "district": "10",
    "street": "Av. 10 de Agosto",
    "description": "Oficina Matriz"
}
```

**Salida:**

```
{
    "succesfull":true/false
}
```

Como parámetros de entrada, el servicio web recibe el identificador de la sucursal del cliente (**branchOfficeld**), las coordenadas de latitud (**latitude**) y longitud (**longitude**) proporcionadas

por el dispositivo móvil e información adicional como la ciudad (**city**), la provincia (**state**) y la parroquia (**district**) proporcionada de forma manual por el vendedor.

También el vendedor ingresa la dirección de la ubicación del cliente (**street**) y una descripción de la misma (**description**).

Si la actualización de la dirección del cliente fue exitosa en el ERP, el servicio web retorna true, caso contrario retorna false.

### INGRESO DE DIRECCIONES (SUCURSALES) DE CLIENTES

El administrador de la aplicación móvil también tiene la capacidad de ingresar una nueva dirección o sucursal de un cliente específico, para cumplir con este objetivo se ha desarrollado este servicio web que cumple con la siguiente estructura:

**Entrada:**

```
{  
  
    "customerId":55455453,  
    "latitude": -0.7736362,  
    "longitude": -17.545443,  
    "state": "17",  
    "city": "UIO",  
    "district": "10"  
    "street": "Av. 10 de Agosto",  
    "description": "Oficina Matriz"
```

```
}
```

**Salida:**

```
{  
  
    "successful":true/false  
  
}
```

Como parámetros de entrada, el servicio web recibe el identificador del cliente (**customerId**), las coordenadas de latitud (**latitude**) y longitud (**longitude**) proporcionadas por el dispositivo móvil e información adicional como la ciudad (**city**), la provincia (**state**) y la parroquia (**district**) proporcionada de forma manual por el vendedor.

También el vendedor ingresa la dirección de la ubicación del cliente (**street**) y una descripción de la misma (**description**).

Si el registro de una nueva dirección del cliente fue exitosa en el ERP, el servicio web retorna true, caso contrario retorna false.

## 5.9 API de Rutas de Google

Una de las funcionalidades que ofrece la aplicación móvil, es sugerir una ruta alternativa desde la ubicación actual del vendedor hacia cualquiera de los clientes (sucursales) que forman parte de la ruta de ventas programada para un día específico.

Esto se lo consigue a través del uso del API de rutas de google, que es un servicio que utiliza una solicitud HTTP que se encarga de calcular la ruta de una ubicación a otra y que recibe como parámetros obligatorios las coordenadas del origen y destino (latitud y longitud).

Una solicitud del API de rutas tiene el siguiente formato:

*<http://maps.googleapis.com/maps/api/directions/salida?parametros>*

El valor **salida** se reemplaza por cualquier de las opciones detalladas a continuación:

- **Json**: indica el formato de salida en Notación de objetos JavaScript (JavaScript Object Notation, JSON).
- **Xml**: indica el formato de salida como un archivo XML.

### PARÁMETROS DE SOLICITUD

Los parámetros que ofrece esta API son múltiples y muy interesantes, sin embargo nos centraremos en aquellos de carácter obligatorio. Como en las URL estándar, todos los parámetros se separan con el carácter **&**. A continuación, se indican los parámetros admitidos y sus posibles valores.

### PARÁMETROS OBLIGATORIOS

- **origin**: permite especificar el valor de latitud/longitud de la ubicación desde la que se desea calcular la ruta.



- **destination:** permite especificar el valor de latitud/longitud de la ubicación hacia la cual nos queremos dirigir.

## PARÁMETROS OPCIONALES

A continuación se menciona ciertos parámetros opciones que vale la pena conocerlos:

- **mode** (el valor predeterminado es driving): especifica el medio de transporte que se utilizará al calcular las posibles rutas. Los medios de transporte disponibles son los siguientes:
  - driving: (predeterminado) proporciona las posibles rutas considerando como medio de transporte el automóvil
  - walking: proporciona las rutas a pie a través de aceras y rutas peatonales
  - bicycling: proporciona las posibles rutas para llegar en bicicleta a través de ciclo vías
  - transit: solicita indicaciones de cómo llegar a un destino especificado a través de rutas de transporte público.
- **alternatives:** si se establece en **true**, indica que el servicio de rutas puede devolver más de una ruta.

## RETORNO DE INFORMACIÓN DE API DE RUTAS

Las respuestas de ruta se devuelven en el formato que indica la marca de **salida** en la ruta de la solicitud de la URL.

Nos centraremos en el formato JSON que corresponde al que utilizamos en el trabajo de titulación propuesto. Para entender de mejor forma el formato de salida, utilizaremos como ejemplo las indicaciones arrojadas por el servicio de rutas de google tomando como origen mi dirección de domicilio (Latitud: -0.158056 y Longitud: -78.480833) y de destino la ubicación de la Plaza de toros Quito.

<http://maps.googleapis.com/maps/api/directions/json?origin=-0.158056,-78.480833&destination=-0.163406,-78.48407417>

```
{
  "routes": [
    {
      "bounds": {
        "northeast": {
          "lat": -0.1581432,
          "lng": -78.48053929999999
        },
        "southwest": {
          "lat": -0.1634842,
          "lng": -78.48350649999999
        }
      },
      "copyrights": "Datos del mapa ©2015 Google",
      "legs": [
        {
          "distance": {
            "text": "0.9 km",
            "value": 864
          },
          "duration": {
            "text": "3 min",
            "value": 203
          },
          "end_address": "Avenida Río Amazonas, Quito 170513, Ecuador",
          "end_location": {
            "lat": -0.1634842,
            "lng": -78.48350649999999
          },
          "start_address": "L.Veloz, Quito 170138, Ecuador",
          "start_location": {
            "lat": -0.1581432,
            "lng": -78.48075249999999
          },
          "steps": [
            {
              "distance": {
                "text": "85 m",
                "value": 85
              },
              "duration": {
                "text": "1 min",
                "value": 11
              },
              "end_location": {
                "lat": -0.1585995,
                "lng": -78.48136290000001
              },
              "html_instructions": "Dirígete al \u003cb\u003esuroeste\u003c/b\u003e por \u003cb\u003ePablo Del Solar\u003c/b\u003e hacia \u003cb\u003eGuepi\u003c/b\u003e.",
              "polyline": {
                "points": "j[]tg_~MzAxB"
              },
              "start_location": {
                "lat": -0.1581432,
                "lng": -78.48075249999999
              },
            }
          ]
        }
      ]
    }
  ]
}
```

```

    "travel_mode" : "DRIVING"
  },
  :
  :
  {
    "distance" : {
      "text" : "0.2 km",
      "value" : 185
    },
    "duration" : {
      "text" : "1 min",
      "value" : 43
    },
    "end_location" : {
      "lat" : -0.1634842,
      "lng" : -78.48350649999999
    },
    "html_instructions" : "Gira a la \u003cb\u003eizquierda\u003c/b\u003e en la 1\u003a intersecci\u003bn hacia \u003cb\u003eAv  
R\u003bo Amazonas\u003c/b\u003e.\u003cddiv style=\"font-size:0.9em\"\u003eEl destino est\u003a a la derecha.\u003c/div\u003e",
    "maneuver" : "turn-left",
    "polyline" : {
      "points" : "Is^jv_~MCVpCVLBTHF@dAJh@D"
    },
    "start_location" : {
      "lat" : -0.161987,
      "lng" : -78.48310359999999
    },
    "travel_mode" : "DRIVING"
  }
],
"via_waypoint" : []
}
],
"overview_polyline" : {
  "points" : "j[[]tg_~MzAxBrAw@|CkBrB~Cx@rA`@j@j@\\^JrDb@[pCCVpCVb@LvBR"
},
"summary" : "Isla Isabela",
"warnings" : [],
"waypoint_order" : []
}
],
"status" : "OK"
}

```

En funci\u003bn de la salida JSON mostrada en la secci\u003bn anterior, explicaremos los objetos usados en el trabajo de titulaci\u003bn propuesto.

- **Legs:** Arreglo que contiene el detalle de todos los tramos (steps) para llegar de un sitio a otro. Si se ha especificado el par\u003ametro **alternatives** en true, existir\u003an tantos **Legs** como rutas alternativas arroje el servicio. Contiene los siguientes objetos que vale la pena mencionar.
  - **Distance:** especifica la distancia en metros desde el punto de origen hacia el destino final.

- **Duration:** especifica el tiempo en minutos desde el punto de origen hacia el destino final.
- **Steps:** Representa cada tramo o punto de quiebre de la ruta.
- **overview\_polyline:** Contiene un objeto que consta de un conjunto de puntos (points) codificados que representan una ruta aproximada (suavizada) de las indicaciones resultantes. Esta información es de carácter relevante para este proyecto, ya que permite representar de forma precisa la ruta en el mapa.

## 5.10 Core Location Framework iOS

La función de localización permite que las aplicaciones móviles basadas en la ubicación (incluyendo Mapas, Cámara, Safari y otras aplicaciones de Apple y de terceros) utilicen información de redes móviles, redes inalámbricas Wi-Fi, el sistema de posicionamiento global (GPS) y iBeacons<sup>4</sup> para determinar la ubicación aproximada.

Dependiendo del dispositivo y de los servicios disponibles, la localización utiliza una combinación de datos móviles, Wi-Fi, Bluetooth y GPS para averiguar la ubicación del dispositivo móvil. Si no se encuentra dentro de la línea de visión de los satélites GPS, el dispositivo podrá determinar su localización utilizando localizaciones masivas de Wi-Fi, torres de telefonía móvil o iBeacons<sup>5</sup>

El marco de trabajo **Core Location** contiene todas las clases necesarias que permiten a las aplicaciones determinar la posición geográfica del dispositivo móvil que corren bajo la plataforma iOS, independientemente del tipo de dispositivo (iPad / iPhone)

Las clases de este marco de trabajo tienen el prefijo CL y para hacer uso de ellas debemos importar esta librería en el código fuente.

```
#import <CoreLocation/CoreLocation.h>
```

---

<sup>4</sup> **iBeacon** es un sistema de posicionamiento en interiores (IPS) basado en bluetooth de baja energía (BLE)

Adicionalmente es necesario declarar un objeto de la clase `CLLocationManager`, esta clase es la encargada de interactuar directamente con el hardware de ubicación del dispositivo móvil.

```
@implementation RouteTableViewController {
```

```
    CLLocationManager *locationManager;
```

```
}
```

Una instancia de la clase *CLLocationManager* tiene un número de propiedades que especifican su comportamiento, dos de estas propiedades son requeridas y detalladas a continuación:

- *distanceFilter*: propiedad que permite determinar la distancia en metros que el dispositivo móvil puede desplazarse antes de que el objeto de la clase *CLLocationManager* notifique a la aplicación que la ubicación ha cambiado.
- *desiredAccuracy*: esta propiedad indica al objeto de la clase *CLLocationManager* cuando precisa debe ser la detección de la ubicación del dispositivo móvil. Esto es importante porque hay una relación directamente proporcional entre la precisión de la localización y el tiempo de vida y consumo de la batería del dispositivo móvil para determinar su ubicación. Por otra parte, la precisión depende también del tipo de dispositivo que el usuario posee, la disponibilidad de antenas de telefonía móvil, satélites y la disponibilidad de puntos de acceso inalámbricos conocidos.

Una vez que estas propiedades han sido configuradas, debemos notificar al objeto de la clase *CLLocationManager* que comience a determinar su ubicación mediante la propiedad *startUpdatingLocation*

A continuación se expone un fragmento del código que muestra todo lo explicado anteriormente.

```
- (void) getCurrentLocation
```

```
{
```

```
    locationManager = [[CLLocationManager alloc] init];
```

```
    locationManager.delegate = self;
```

```
    locationManager.desiredAccuracy = kCLLocationAccuracyBest;
```

```
[locationManager startUpdatingLocation];  
}
```

Para obtener la ubicación actual, primero lo almacenamos en una variable tipo CLLocation y hacemos uso de uno de los protocolos que ofrece la clase CLLocationManager denominado didUpdateLocations.

**@implementation RouteTableViewController {**

```
    CLLocationManager *locationManager;  
  
    CLLocation *currentLocation;  
}  
  
- (void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray  
*)locations {  
  
    currentLocation = [manager location];  
  
    [manager stopUpdatingLocation];  
}
```

En el fragmento de código anterior, la variable currentLocation almacena el valor actual de la posición geográfica mediante la llamada al método location y el método stopUpdatingLocation solicita al hardware de ubicación del dispositivo móvil la cancelación de la determinación de su ubicación.

Para acceder a las coordenadas de latitud y longitud de la posición geográfica actual, basta con acceder a las propiedades de la variable currentLocation tal como se muestra a continuación:

Longitud: currentLocation.coordinate.longitude

Latitud: currentLocation.coordinate.latitude

Para ilustrar cualquier tipo de información sobre un mapa hemos hecho uso de la librería de google maps, Las clases de esta librería tienen el prefijo GM y para hacer uso de ella debemos importarla en el código fuente:

**#import <GoogleMaps/GoogleMaps.h>**

A continuación se incluye un fragmento del código de la aplicación móvil que permite mostrar en el mapa la ruta sugerida desde la ubicación actual del vendedor hacia un determinado cliente.

```
GMSCameraPosition *camera1 = [GMSCameraPosition  
cameraWithLatitude:self.currentLocation.coordinate.latitude  
longitude:self.currentLocation.coordinate.longitude  
zoom:14];  
  
GMSMapView *mapView = [GMSMapView mapWithFrame:CGRectZero camera:camera1];  
  
GMSMarker *originMarker = [[GMSMarker alloc] init];  
originMarker.position = camera1.target  
originMarker.snippet = @"Ubicación Actual";  
originMarker.appearAnimation = kGMSMarkerAnimationPop;  
originMarker.map = mapView;
```

Se ha declarado un objeto (camera1) de la clase GMSCameraPosition, que permite ubicar el foco en una posición determinada del mapa, que para nuestro caso corresponde a la ubicación actual del vendedor. La inicialización de este objeto es necesaria para instanciar al objeto mapView de la clase GMSMapView que es la encargada de presentar el mapa.

El objeto originMarker de la clase GMSMarker permite dibujar en el mapa los denominados marcadores de posición, y tiene propiedades que permiten ubicarla en una posición específica (position), mostrar un texto al hacer clic sobre el marcado (snippet)

Para ubicar la ubicación del cliente, hacemos uso de un segundo marcador (destinationMarker), tal como se muestra a continuación:

```
GMSMarker *destinationMarker = [[GMSMarker alloc] init];  
destinationMarker.position =  
CLLocationCoordinate2DMake(self.destinationLocation.coordinate.latitude,  
self.destinationLocation.coordinate.longitude);  
destinationMarker.snippet = @"Destino Cliente";  
destinationMarker.appearAnimation = kGMSMarkerAnimationPop;  
destinationMarker.map = mapView;
```

Es importante mencionar que en este caso, la propiedad "position" es inicializada con las coordenadas geográficas de la ubicación del cliente.

Para trazar la ruta entre el origen y el destino, hacemos uso del objeto “overview\_polyline” retornado por el API de rutas de Google, cuyo valor está almacenado en la variable pathPolyline.

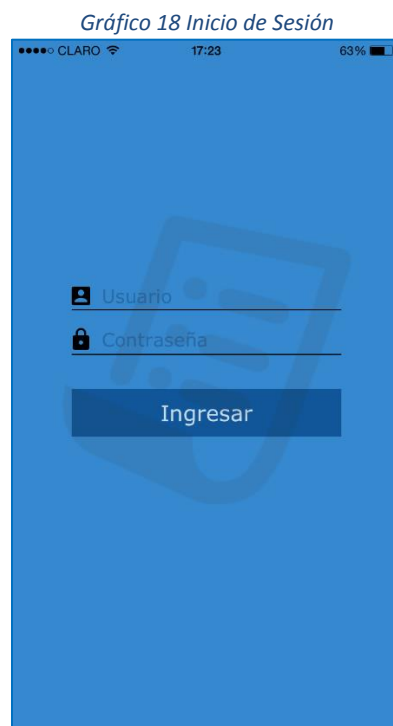
```
GMSPolyline *polyline = [GMSPolyline polylineWithPath:pathPolyline];  
polyline.strokeColor = [UIColor colorWithRed:67.0/255.0 green:169.0/255.0 blue:255.0/255.0  
alpha:1];  
polyline.strokeWidth = 5.f;  
polyline.map = mapView;
```

Finalmente para mostrar en pantalla el mapa con la ruta sugerida, hacemos uso del siguiente comando:

```
self.view = mapView;
```

### 5.11 Aplicación Móvil

La aplicación móvil cuenta con una pantalla de inicio de sesión, donde el vendedor deberá ingresar las credenciales proporcionadas por el administrador del módulo de gestión de rutas de ventas en el sistema central.

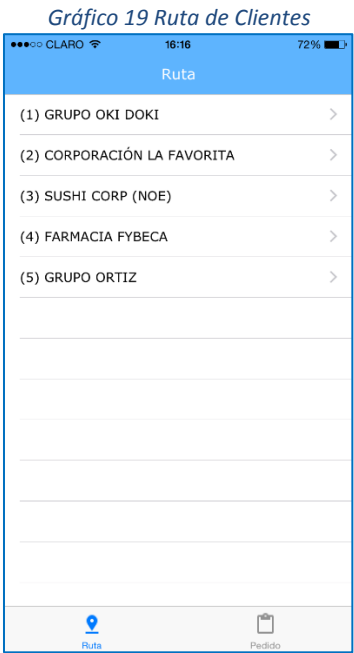


Fuente: Aplicación Móvil

Si se trata de un usuario cuyo perfil es vendedor, se desplegará un listado con todos los clientes que se encuentran en la planificación de visitas para un día específico.



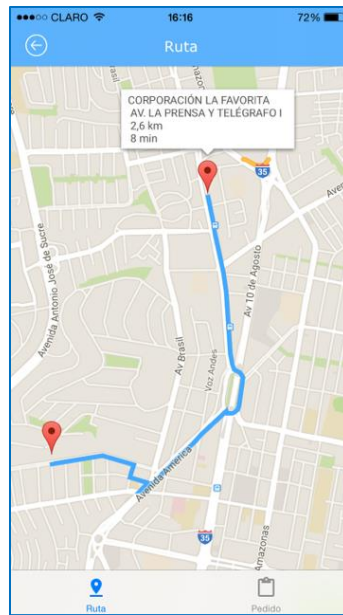
Los clientes se despliegan ordenados ascendentemente en función de los más cercanos, tomando como punto de origen la ubicación del vendedor.



Fuente: Aplicación Móvil

Al hacer clic en cualquier de los clientes, se despliega un mapo que identifica claramente los puntos de origen que corresponde a la ubicación actual del vendedor y el destino que corresponde a la dirección del cliente a visitar; en función de esta información la aplicación dibujará en el mapa una ruta sugerida.

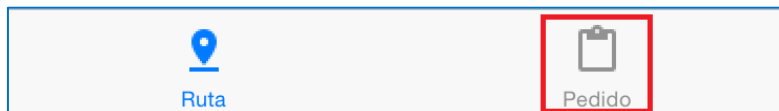
Gráfico 20 Ruta Sugerida - Aplicación Móvil



Fuente: Aplicación Móvil

Para generar un pedido, seleccionamos la opción “Pedido” ubicado en la barra de tareas de la aplicación.

Gráfico 21 Generación de Orden de Venta



Fuente: Aplicación Móvil

En esta nueva pantalla se despliega todas las órdenes de venta generadas por el vendedor durante el día.

Gráfico 22 Órdenes de venta generadas

Pedido	
FARMACIA FYBECA	
Subtotal	
Iva	
<b>TOTAL</b>	
Cliente	FARMACIA FYBECA
RUC	1708042662001
Dirección	AV. CORUÑA Y FRANCISCO DE O...
Detalle	
REP.CACAO EXPL...	1
CHOCOTIN CRIS...	1
GLOBOOM BOTE...	1
BOOGIE MINICHI...	1

Fuente: Aplicación Móvil

Para generar una nueva orden de venta (pedido de venta), hacemos clic en la opción “+”.

Gráfico 23 Crear Orden de Venta

Fuente: Aplicación Móvil

A continuación se despliega una pantalla para permitir el ingreso de una nueva orden de venta como se puede observar en el gráfico No. 24.

Gráfico 24 Creación de Orden de Venta

CLARO 19:15 51%

← Pedido Ordenar

Subtotal	0
Iva	0
<b>TOTAL</b>	<b>0</b>

Cliente Selecciona... >

RUC ...

Dirección ...

Detalle +

Ruta Pedido

Fuente: Aplicación Móvil

Durante el proceso de creación del pedido de venta, es necesario seleccionar el cliente al cual vamos a facturar. Para ello es necesario hacer clic en la opción “*Selecciona*”.

Gráfico 25 Creación de Orden de Venta

CLARO 19:21 50%

← Pedido Ordenar

Subtotal	0
Iva	0
<b>TOTAL</b>	<b>0</b>

Cliente SUSHI CORP (NOE) >

RUC 1715960173001

Dirección PEDRO DEL SOLAR E4-49 Y GUEPI

Detalle +

Ruta Pedido

Fuente: Aplicación Móvil

Una vez seleccionado el cliente, en la sección de detalle de la orden de venta seleccionamos el artículo o artículos a entregar al cliente.

Gráfico 26 Detalle de Orden de Venta

CLARO 19:34 48%

← Pedido Ordenar

Subtotal	1750.90
Iva	210.11
<b>TOTAL</b>	<b>1961.01</b>

Cliente SUSHI CORP (NOE) >

RUC 1715960173001

Dirección PEDRO DEL SOLAR E4-49 Y GUEPI

Detalle +

AMERICAN TOFF...	50	- +
REP.CACAO EXPL...	30	- +
CHOCOTIN CRIS...	100	- +
BOOGIE MINICH...	30	- +
PASTILLA VENDI...	50	- +

Ruta Pedido

Fuente: Aplicación Móvil

Para registrar la orden de venta, se debe hacer clic sobre la opción “Ordenar”, y si el proceso fue correcto se debe mostrar un mensaje como el que se puede observar en el gráfico 27:

Gráfico 27 Factura de Orden de Venta

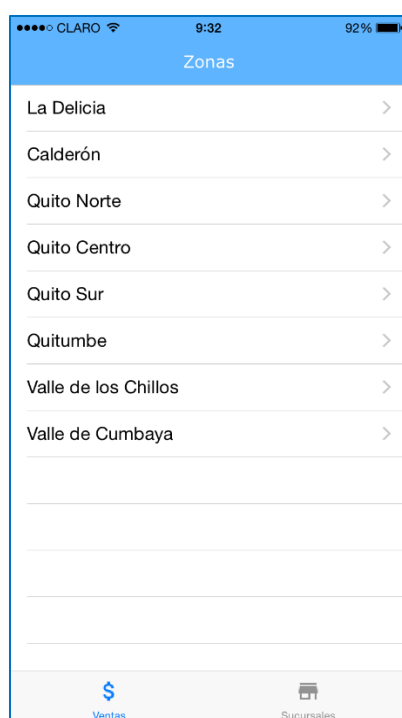


Fuente: Aplicación Móvil

El usuario con perfil administrador, tiene la posibilidad de monitorear las actividades de cada vendedor, esto significa que puede verificar las ventas realizadas por cada vendedor, los clientes que tiene pendiente de visitar y si la venta fue registrada en el sitio de ubicación del cliente.

El proceso de búsqueda inicia en función de la división de zonas indicada en la tabla 3

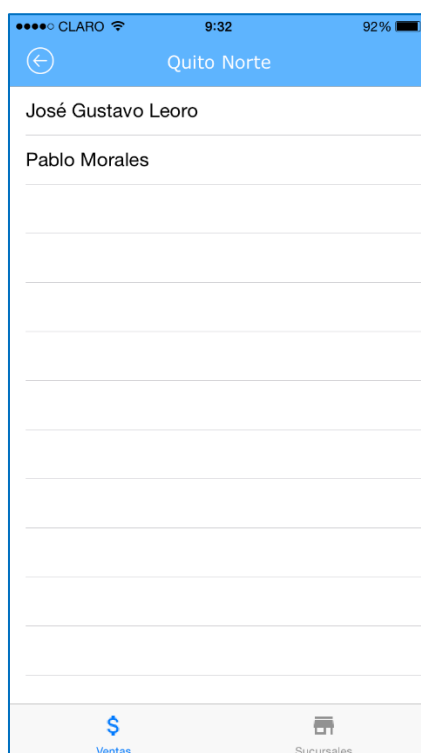
Gráfico 28 Zonas de Venta



Fuente: Aplicación Móvil

. Al ingresar a cada zona de venta se desplegarán los vendedores pertenecientes a dicha zona.

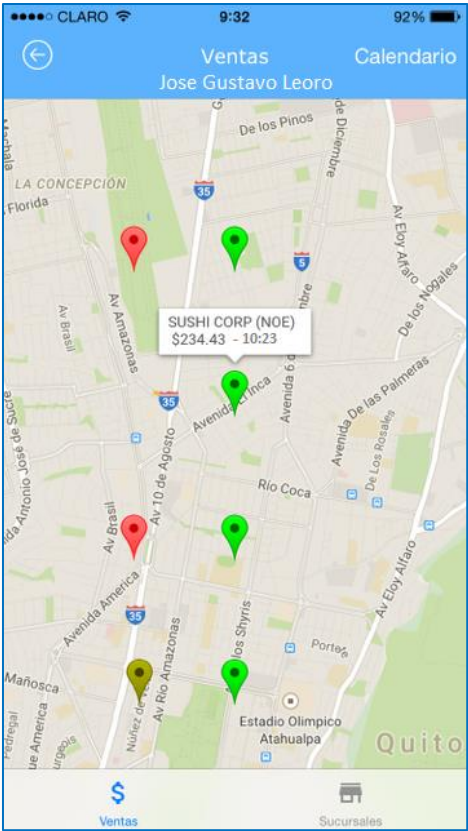
Gráfico 29 Listado de Vendedores



Fuente: Aplicación Móvil

Al seleccionar un vendedor en particular, se despliega la ruta del vendedor, donde se podrá conocer el valor facturado a cada cliente, los clientes pendientes de facturar y distinguir claramente si el vendedor realizó el registro de venta desde la ubicación del cliente para efectos de control.



Gráfico 30 Monitoreo de actividades - Vendedor




Fuente: Aplicación Móvil

Como se puede observar en el gráfico No. 30, se distinguen marcadores con 3 colores diferentes. Cada color tiene su significado el cual se detalla en la tabla No.5:

Tabla 5 Indicador de actividad del vendedor

MARCADOR	DESCRIPCIÓN
	Los marcadores de color verde corresponden a los clientes que ya fueron visitados por el vendedor y cuya factura ya fue registrada en Dynamics Ax
	Los marcadores de color rojo corresponden los clientes que están pendientes de facturar



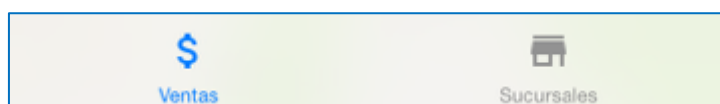
	<p>Los marcadores de color amarillo corresponden a los clientes cuya factura ya fue registrada en Dynamics Ax, pero no fue registrada desde la ubicación del cliente.</p>
---	---

Fuente: Aplicación Móvil

Al seleccionar la opción *Calendario* (mirar gráfico 30), podremos consultar el registro de actividades del vendedor para un día en particular.

Por otra parte, para facilitar el registro de las sucursales de los clientes, el administrador de la aplicación móvil tiene la capacidad de registrar nuevas sucursales o editar una sucursal existente.

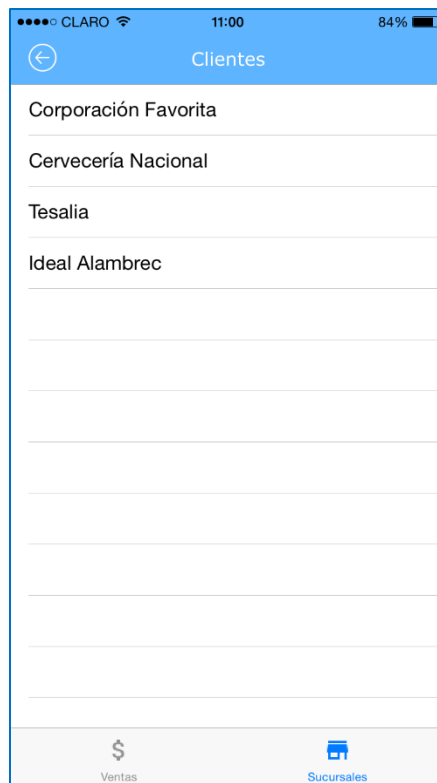
Gráfico 31 Registro de Sucursales



Fuente: Aplicación Móvil

Esta actividad se lo realiza haciendo clic sobre la opción *Sucursales*. En esta opción se desplegará el listado de clientes registrados en el sistema ERP.

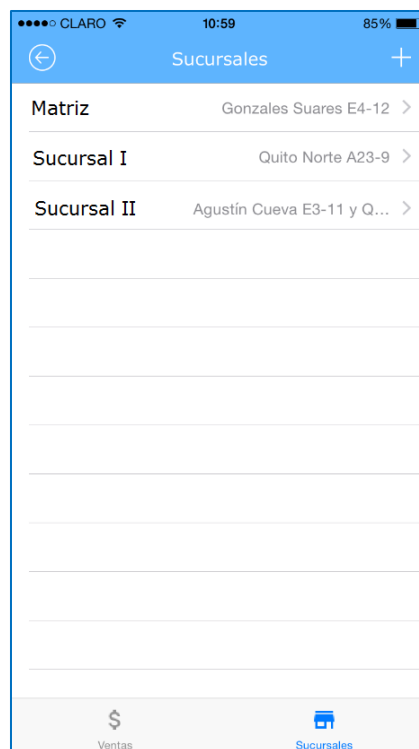
Gráfico 32 Listado de Clientes



Fuente: Aplicación Móvil

Al seleccionar un cliente, se despliegan todas las sucursales asociadas a dicho cliente y que se encuentran registradas en el ERP.

Gráfico 33 Listado de Sucursales por Cliente



Fuente: Aplicación Móvil

Para registrar una nueva sucursal hacemos clic en la opción “+” y para editar una sucursal existente basta con seleccionar la sucursal haciendo clic sobre ella.

Para el registro o edición de una sucursal, se despliega la siguiente pantalla.

*Gráfico 34 Ingreso o edición de sucursales*

SUSHI CORP (NOE)

SUCURSAL I

PEDRO DEL SOLAR E4-49 Y GUEPI

Provincia PICHINCHA >

Ciudad QUITO >

Parroquia JIPIJAPA >

Reemplazar Ubicación ☐

Ubicación Actual ( -0.170619, -78.495483 )

Cientes Ventas Ruta Pedido

Fuente: Aplicación Móvil

En la tabla No. 6 se detalla cada elemento del a pantalla mostrada en el gráfico 34.

*Tabla 6 Registro de Sucursales*

Campo	Descripción
Sucursal	Una descripción general de la dirección del cliente
Calle	Ubicación del cliente
Provincia	Corresponde a la provincia en la cual se ubica el cliente.
Ciudad	Corresponde a la ciudad en la cual se ubica el cliente.
Parroquia	Corresponde a la parroquia en la que se ubica el cliente.
Localización Geográfica	Al activar la opción de reemplazar ubicación, se actualiza las coordenadas de latitud y longitud tomando como referencia la ubicación del dispositivo móvil

Fuente: Aplicación Móvil

Al hacer clic en la opción “*Guardar*” se registra toda la información relacionada con la sucursal, salvo la localización geográfica. Para poder registrar esta información es necesario hacer clic sobre el ícono del mapa.

## 5.12 Funcionamiento Off-Line

Para aquellos casos en que se tenga una cobertura de señal limitada o ausencia total de señal, la aplicación móvil permite trabajar en modo Off-line, con la finalidad de que el proceso de registro de pedidos de venta no se interrumpa.

Para lograr este objetivo, la información de la ruta del vendedor, que comprende el stock de inventario y el conjunto de clientes a visitar, se almacena en una base de datos interna de aplicación y es actualizada durante el primer inicio de sesión del vendedor en el día; esto significa que para el primer inicio de sesión, el vendedor necesariamente deberá contar con cobertura de datos móviles en su dispositivo celular.

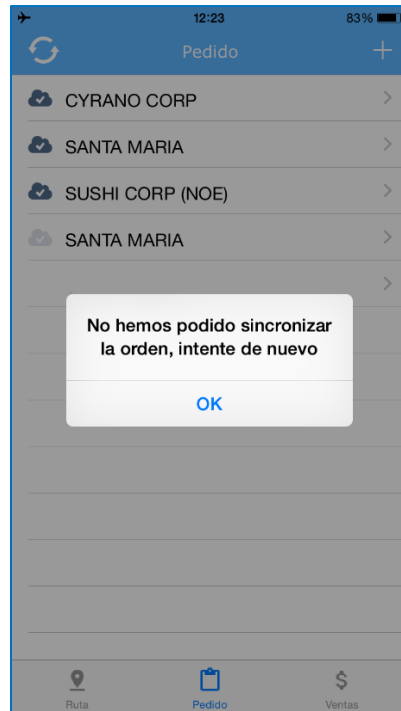
En el caso de iOS, este cuenta con un marco de trabajo para el manejo y almacenamiento de datos de una aplicación conocido como Core Data, que permite realizar las operaciones básicas CRUD<sup>6</sup> sobre una base de datos relacional almacenada en SQLite. Core Data es un ORM (mapeo de objeto-relacional) que permite convertir los objetos del lenguaje de programación Objective-C en datos que son almacenados en una base de datos SQLite y viceversa; en términos más sencillos, este ORM brinda la capacidad de extraer y almacenar datos en una base de datos relacional sin la necesidad de conocer el lenguaje SQL. La estructura de la base interna se lo puede mirar en el anexo 3 de este documento.

Cuando la aplicación no detecta cobertura de señal o no existe conectividad con el servidor central, el vendedor es notificado de dicho inconveniente al momento de registrar el pedido de venta.

---

<sup>6</sup> Término usado para referirse a las operaciones básicas ejecutadas sobre una base de datos: Insertar (create), leer (read), actualizar (update) y borrar (delete)

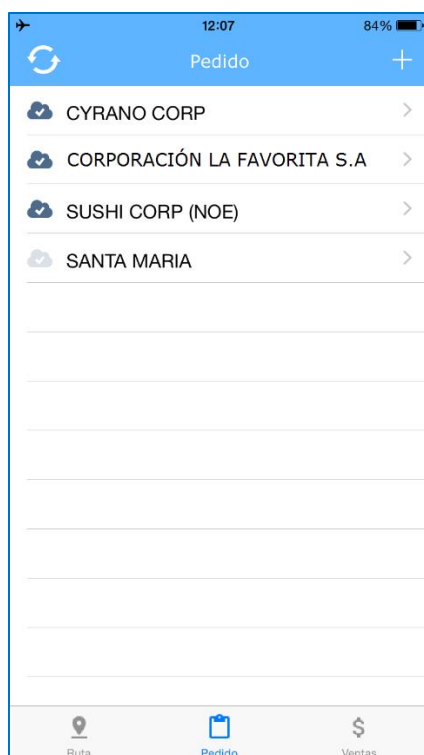
Gráfico 35 Sincronización de Pedidos de Venta




Fuente: Aplicación Móvil

Los pedidos que se encuentran pendientes de sincronizar son identificados claramente como se muestra en el gráfico No. 36.

Gráfico 36 Pedidos Pendientes de Sincronizar



Fuente: Aplicación Móvil

Cuando el vendedor disponga nuevamente de cobertura, podrá realizar la sincronización de sus pedidos haciendo clic en el botón. 

Es importante mencionar que las tareas administrativas de la aplicación relacionadas con el monitoreo de actividades de cada vendedor y administración de sucursales de clientes requieren necesariamente de conectividad móvil 3G/E o red inalámbrica.

## 5.13 Conclusiones y Recomendaciones

### CONCLUSIONES

- El desarrollo del módulo de gestión y administración de rutas fue concebido tomando como ejemplo el modelo de ventas de una de las empresas ecuatorianas líderes en la comercialización y distribución de confites, con el objetivo de convertirlo en una base sólida que pueda acoplarse y personalizarse a otras empresas de similares características y que en primera instancia tengan como sistema central a Microsoft Dynamics Ax, que son alrededor de más de 40 empresas a nivel nacional.
- La aplicación móvil ha sido desarrollada completamente bajo Objective-C, que es el lenguaje de programación para las plataformas MAC OS X y iOS, si bien está considerado como el cuarto lenguaje más popular a nivel mundial<sup>7</sup>, su curva de aprendizaje y complejidad es alto, tanto desde el punto de vista sintáctico como también en el proceso de inicialización o construcción de objetos, por eso es que Apple está volcando sus esfuerzos al desarrollo del nuevo lenguaje bautizado como Swift.
- Microsoft Dynamics Ax permite exponer cualquier funcionalidad desarrollada bajo X++ hacia el exterior mediante el uso de servicios web, para ello dispone de un asistente que facilita esta operación generando las clases necesarias para cumplir con este objetivo, sin embargo el uso de este asistente hace que la modificación posterior de estas clases y su mantenimiento sea mucho más complejo, el uso de clases personalizadas para exponer código X++ como servicios web permiten tener un mayor control sobre el código fuente y un mejor control de cambios.
- El intercambio de información entre el sistema central y la aplicación móvil es a través del uso de servicios web. El esquema seleccionado REST-JSON permite que la información a intercambiar se reduzca en cuanto a tamaño en comparación con el estándar SOAP-XML, que a su vez se traduce en el envío de menor cantidad de información sobre la red móvil, lo que es de vital importancia en el diseño de aplicaciones móviles.

---

<sup>7</sup> <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>



- La arquitectura REST-JSON, es mucho más simple de analizarla sintácticamente en comparación con el formato XML, lo que hace que su extracción y conversión (deserialización) sea más simple, obteniéndose menores tiempos de uso de CPU y procesamiento en los dispositivos móviles.
- Una buena práctica en el tema de desarrollo de aplicaciones y más aún cuando el equipo de desarrollo está conformado por más de una persona, es utilizar herramientas que permitan manejar el control de versiones y cambios, para el desarrollo de la aplicación móvil se utilizó el repositorio de código fuente gratuito denominado Bitbucket<sup>8</sup> que posee una interfaz muy intuitiva y permite llevar a cabo esta actividad con facilidad, actualmente soporta varios lenguajes de programación que aplican para el desarrollo de aplicaciones sobre iOS y Android.
- Para la detección de errores en cualquier aplicación móvil, específicamente durante caídas inesperadas de la aplicación, existen una serie de librerías como Crashlytics que se encargan de notificar al desarrollador (vía correo electrónico) sobre cualquier fallo excepción de la aplicación, notificando información valiosa como la fecha y la hora de la excepción y la línea de código fuente donde se produjo el error; fue de vital ayuda en la construcción del prototipo ya que permitió identificar fallos importantes en la diagramación de los mapas.
- El disponer de información en línea, para procesos cruciales dentro de cualquier organización como es la gestión de ventas, y permitir el monitoreo de las actividades del personal de ventas, junto con una simplificación en el ingreso de pedidos de venta, hacen del prototipo propuesto una herramienta ideal para toda aquella organización de distribución masiva de productos, además de que resulta intuitiva en su manejo lo que reducirá considerablemente los tiempos de capacitación al usuario final.
- Actualmente un importante número de empresas ecuatorianas han optado por integrar su información, dejando atrás los múltiples sistemas orientados a funciones o procesos específicos muy puntuales dentro de una organización y han migrado a sistemas más robustos y confiables como un ERP, añadir características de movilidad a este tipo de sistemas en los procesos cruciales dentro de una organización sin lugar a duda representa una ventaja competitiva sobre el resto de empresas.

---

<sup>8</sup> <https://bitbucket.org/>

## RECOMENDACIONES

- Para agilizar el proceso de despacho durante de la entrega de los productos hacia el cliente, y tomando en cuenta que en muchos casos el volumen de productos a despachar puede ser considerable, se recomienda incluir el manejo de código de barras, de esta manera el ingreso del detalle del pedido de venta (productos) se lo hará a través de la lectura del código de barras mediante el uso del lente de la cámara del dispositivo móvil.
- Contar en el mediano plazo con una versión disponible para dispositivos móviles basados en sistemas operativos Android y Windows Phone, con el objetivo de hacerlo atractivo a un mayor número de clientes que poseen Dynamics Ax a nivel nacional e internacional
- El desarrollo de este tipo de aplicaciones requiere un conocimiento diverso en cuanto a lenguajes de programación, en el caso del proyecto propuesto, esto significa tener un amplio conocimiento del lenguaje de programación x++ para realizar modificaciones sobre Microsoft Dynamics Ax, C# o cualquiera de los lenguajes soportados por .NET para el desarrollo de los servicios web y Objective-C para el desarrollo de la aplicación móvil. Para atenuar la complejidad, ya existe en el mercado propuestas interesantes como XAMARIN que permite utilizar C# y .NET para el desarrollo de aplicaciones móviles sobre Android y iOS, por lo tanto se recomienda profundizar en el conocimiento de este tipo de plataformas y aplicarlo sobre todo para aquellos desarrolladores que están completamente familiarizados con las herramientas de desarrollo Microsoft.
- Dentro del ámbito logístico, un producto puede tener atributos o características, en Dynamics Ax esto se conoce como dimensiones de producto, actualmente el manejo de las dimensiones de producto no ha sido considerado como parte del prototipo inicial, sin embargo se recomienda incluir estas características en versiones posteriores.
- Se recomienda que todo intercambio de información a través de servicios web se lo realiza sobre canales seguros que provean autenticación segura, confidencialidad e integridad.

## VI. Bibliografía:

- Microsoft TechNet. (2012). *Acerca de los sitios y la funcionalidad de multisitio [Ax 2012]* Recuperado el 15 de Abril de 2015, de <https://technet.microsoft.com/es-es/library/dd309972.aspx>
- Microsoft TechNet. (2013). *Services and AIF architecture [Ax 2012]*. Recuperado el 15 de Abril de 2015, de <https://technet.microsoft.com/en-us/library/dd309705.aspx>
- Microsoft Developer Network. (2013). *Windows Communication Foundation*. Recuperado el 15 de Abril de 2015, de <https://msdn.microsoft.com/es-es/library/dd456779.aspx>
- Microsoft Developer Network. (2013). *¿Qué es Windows Communication Foundation?*. Recuperado el 15 de Abril de 2015, de <https://msdn.microsoft.com/es-es/library/ms731082%28v=vs.110%29.aspx>
- Microsoft TechNet. (2013). *Using custom services [AX 2012]*. Recuperado el 15 de Abril de 2015, de <https://technet.microsoft.com/en-us/library/hh509052.aspx>
- Joe Conway & Aaron Hillegass, iOS Programming The Big Nerd Ranch Guide 2da Edición, Delegation and Core Location, pagina 77-81
- Microsoft Corporation, Microsoft Dynamics AX 2012 Installation Guide, Enero 2015
- Ecma International. (2013). *Standard ECMA-404: The JSON Data Interchange Format*. Recuperado el 16 de Abril de 2015, de <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>
- Google Developers. (2013). *El API de rutas de Google*. Recuperado el 1 de Abril de 2015, de <https://developers.google.com/maps/documentation/directions/>
- Apple Corporation. (2014). *iOS: Como funciona la localización*. Recuperado el 4 de Abril de 2015, de <https://support.apple.com/es-es/HT5594>
- Microsoft TechNet. (2011). *Security architecture for Web services [AX 2012]*. Recuperado el 16 de Mayo de 2015, de <https://technet.microsoft.com/EN-US/library/dd309651.aspx>
- Microsoft TechNet. (2013). *Integration Ports [AX 2012]*. Recuperado el 16 de Mayo de 2015, de <https://technet.microsoft.com/en-us/library/gg731821.aspx>
- Apple Developer. (2013). *Core Location Framework Reference*. Recuperado el 4 de Abril de 2015, de [https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CoreLocation\\_Framework/](https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CoreLocation_Framework/)

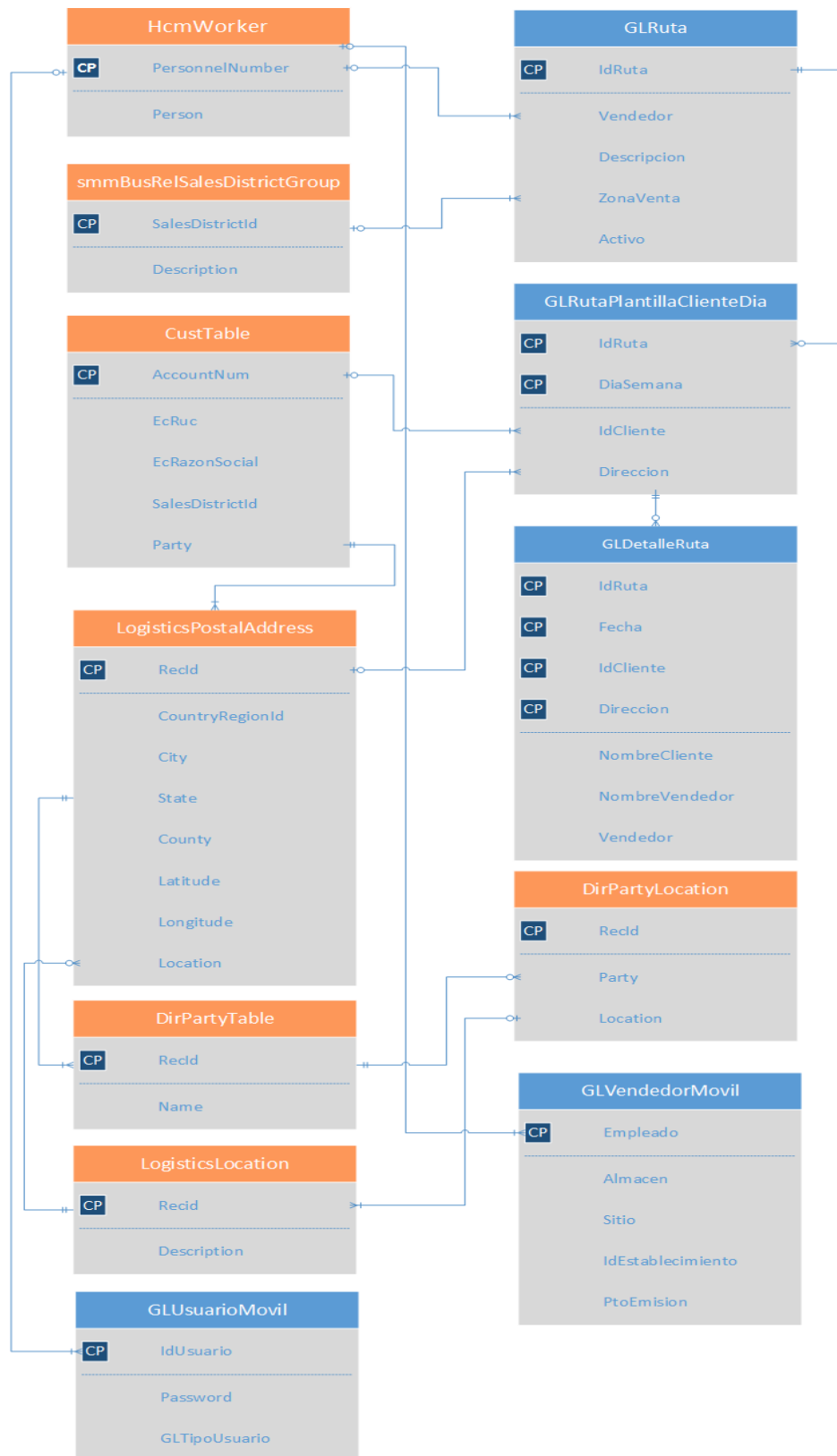
- Google Developers. (2013). *Google Maps SDK for iOS*. Recuperado el 1 de Abril de 2015, de <https://developers.google.com/maps/documentation/ios/>

## **VII. Anexos**

### **ANEXO 1. DIAGRAMA ENTIDAD RELACIÓN:**

A continuación se adjunta el diagrama entidad relación del módulo de administración de rutas de venta desarrollado bajo Microsoft Dynamics Ax.

Gráfico 37 Diagrama Entidad Relación - Módulo de gestión de rutas



Fuente: Gustavo Leoro

## Anexo 2. DICCIONARIO DE DATOS

- **Tabla GLUsuarioMovil:** Almacena la información de credenciales y perfiles de los vendedores que tendrán acceso a la aplicación móvil.

*Tabla 7 Diccionario de datos - Usuarios Móviles*

CAMPO	TIPO DE DATO	OBSERVACIÓN
IdUsuario	String (25)	Representa al nombre de usuario del vendedor, que corresponde al mismo código de empleado en el ERP
Password	String (10)	Corresponde a la contraseña de acceso de cada vendedor para acceder a la aplicación móvil
GLTipoUsuario	Int (enumeración)	Identifica el rol que desempeñara el vendedor dentro de la aplicación móvil. Existen dos perfiles predeterminados:  0 = Administrador 1 = Vendedor

Fuente: Dynamics Ax – Módulo de Gestión de Rutas

- **Tabla GLVendedorMovil:** Almacena la información de credenciales y perfiles de los vendedores que tendrán acceso a la aplicación móvil.

*Tabla 8 Diccionario de Datos - Información adicional de vendedores*

CAMPO	TIPO DE DATO	OBSERVACIÓN
Empleado	String (25)	Representa el código identificador del empleado (vendedor)
Sitio	String (10)	Almacena el sitio atado al vendedor. Un sitio corresponde a una agrupación de recursos empresariales como por ejemplo almacenes y poseen una ubicación geográfica.
Almacen	String (10)	Especifica la bodega atada al vendedor, una bodega corresponde a una ubicación donde se almacena todo tipo de producto distribuido y/o producido por la organización.
idEstablecimiento	String (3)	Especifica el establecimiento atado al vendedor. Esta información es requerida para facturar el pedido de venta creado en el proceso de despacho
PtoEmision	String (3)	Especifica el punto de emisión atado al vendedor. Esta información es requerida para facturar el pedido de venta creado en el proceso de despacho

Fuente: Dynamics Ax – Módulo de Gestión de Rutas

- **Tabla GLRuta:** Almacena la información necesaria para identificar a una ruta, la cual se detalla en la tabla No.9:

*Tabla 9 Diccionario de Datos - Información general de Ruta*

CAMPO	TIPO DE DATO	OBSERVACIÓN
idRuta	String (15)	Representa el código identificador de la ruta.
Vendedor	String (25)	Almacena el código del vendedor atado a una ruta específica

Descripción	String (10)	Permite almacenar una descripción de la ruta de venta.
ZonaVenta	String (3)	Permite atar a una ruta con una zona de venta.
Activo	Int	Permite identificar si una ruta esta activa o no. Únicamente las rutas activas son consideradas en el envío de información hacia el dispositivo móvil. 0 = No 1 = Si

Fuente: Dynamics Ax – Módulo de Gestión de Rutas

- **Tabla *GLRutaPlantillaClienteDia*:** Almacena el detalle de información de la ruta. Esta tabla permite especificar para cada día de la semana, la ruta de ventas para cada vendedor. Actúa como una plantilla en función de la cual se arma la planificación de rutas de venta para un periodo de tiempo especificado. Está compuesta por los siguientes campos:

Tabla 10 Diccionario de Datos - Plantilla de Rutas de Venta

CAMPO	TIPO DE DATO	OBSERVACIÓN
idRuta	String (15)	Representa el código identificador de la ruta
DiaSemana	Int	Especifica el día de la semana.  1 = Lunes 2 = Martes 3 = Miércoles 4 = Jueves 5 = Viernes 6 = Sábado 7 = Domingo
IdCliente	String (20)	Almacena el código del cliente a visitar
Direccion	Int64	Almacena el identificador de la dirección de la sucursal del cliente a visitar.

Fuente: Dynamics Ax – Módulo de Gestión de Rutas

- **Tabla *GLDetalleRuta*:** Una vez que se corre la programación de rutas de venta en un rango de tiempo, esta tabla almacena el detalle de las rutas de venta final para cada vendedor, tomando siempre como información base la plantilla de rutas de venta (tabla ***GLRutaPlantillaClienteDia***).

Tabla 11 Información de Detalle de Ruta

CAMPO	TIPO DE DATO	OBSERVACIÓN
idRuta	String (15)	Representa el código identificador de la ruta.
Vendedor	String (25)	Almacena el código del vendedor atado a una ruta específica
Fecha	Date	Almacena la fecha de programación de cada visita.
Direccion	Int64	Almacena el identificador de la dirección de la sucursal del cliente a visitar.
IdCliente	String (20)	Almacena el identificador del cliente a visitar
NombreCliente	String (100)	Almacena el nombre del cliente a visitar
NombreVendedor	String (100)	Almacena el nombre del vendedor atado a la ruta de venta

Fuente: Dynamics Ax – Módulo de Gestión de Rutas



Adicional, se realizaron modificaciones sobre algunas tablas estándares del sistema para acoplar de mejor manera la solución propuesta en el presente proyecto.

- **Tabla SalesTable:** Esta tabla almacena la información de cabecera de una orden de venta. Se ha realizado modificaciones para guardar la información de la ubicación geográfica donde se registra el pedido de venta junto con el identificador de ruta.

*Tabla 12 Diccionario de Datos - Información de Orden de Venta*

CAMPO	TIPO DE DATO	OBSERVACIÓN
GLIdRuta	String (15)	Almacena el identificador de ruta.
GLLatitude	Float	Almacena la posición geográfica (latitud) donde se registra el pedido de venta.
GLLongitude	Float	Almacena la posición geográfica (longitud) donde se registra el pedido de venta.

Fuente: Dynamics Ax – Módulo de Gestión de Rutas

- **Tabla CustInvoiceJour:** Esta tabla almacena todas las facturas dirigidas a clientes generadas en la organización en el módulo de cuentas por cobrar de Microsoft Dynamics Ax. Se realizaron modificaciones sobre esta tabla para guardar la información de la ubicación geográfica donde se registra la factura del pedido de venta, junto con el identificador de ruta y el vendedor.

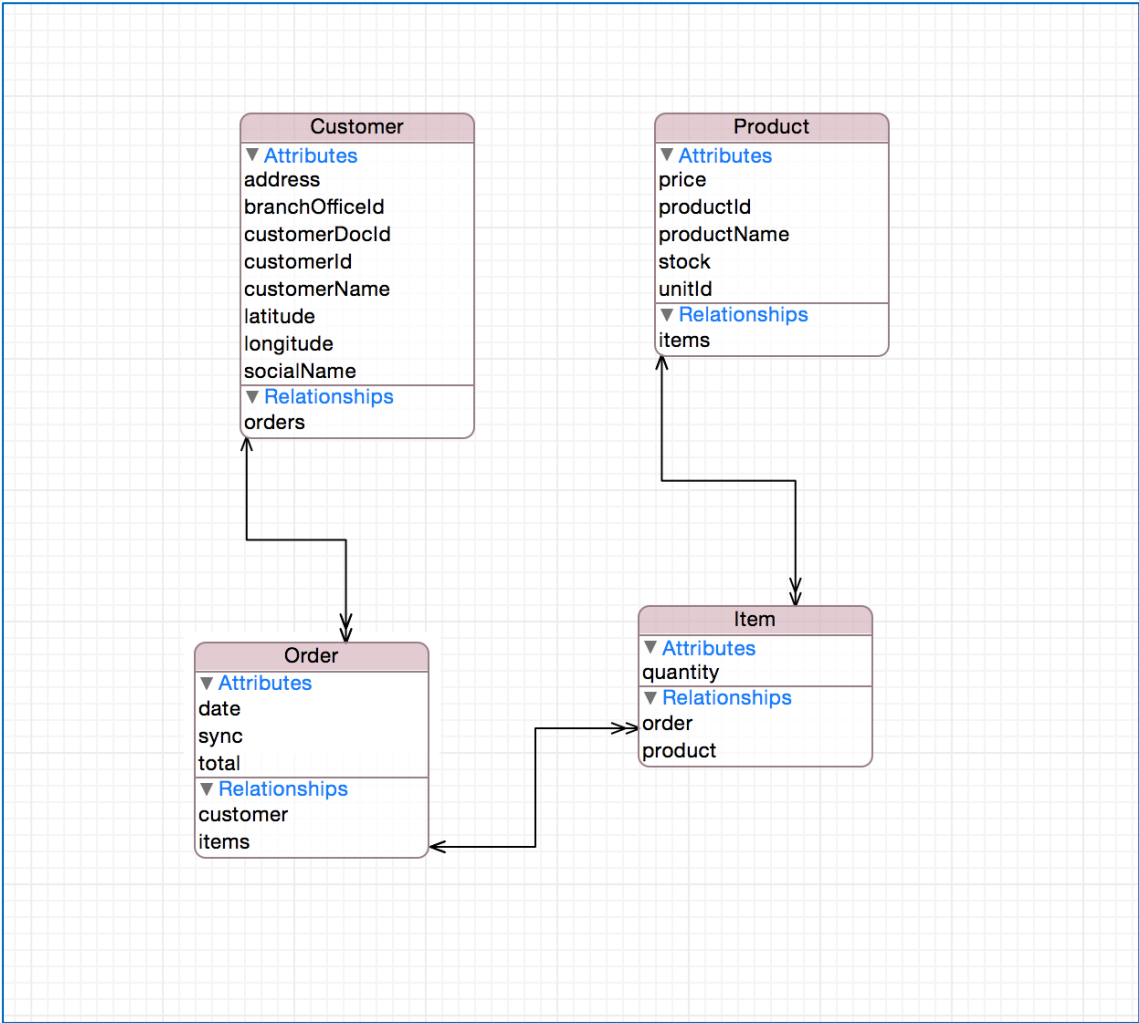
*Tabla 13 Diccionario de Datos - Información de Facturas de Orden de Venta*

CAMPO	TIPO DE DATO	OBSERVACIÓN
GLIdRuta	String (15)	Almacena el identificador de ruta.
GLLatitude	Float	Almacena la posición geográfica (latitud) donde se registra la factura de venta
GLLongitude	Float	Almacena la posición geográfica (longitud) donde se registra la factura de venta
GLVendedor	String (25)	Almacena el código del vendedor que ejecutó el registro de la factura de venta

Fuente: Dynamics Ax – Módulo de Gestión de Rutas

Anexo 3. Diagrama de clases iOS (Core Data Model)

Gráfico 38 Diagrama de clases de la aplicación móvil



Fuente: Aplicación Móvil